A Meshless Method for Variational Nonrigid 2-D Shape Registration

Wei Liu and Eraldo Ribeiro

Computer Vision and Bio-Inspired Computing Laboratory
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, U.S.A.
eribeiro@cs.fit.edu

Abstract

We present a method for nonrigid registration of 2-D geometric shapes. Our contribution is twofold. First, we extend the classic chamfer-matching energy to a variational functional. Secondly, we introduce a meshless deformation model that can handle significant high-curvature deformations. We represent 2-D shapes implicitly using distance transforms, and registration error is defined based on the shape contours' mutual distances. In addition, we model global shape deformation as an approximation blended from local deformation fields using partition-of-unity. The global deformation field is regularized by penalizing inconsistencies between local fields. The representation can be made adaptive to shape's contour, leading to registration that is both flexible and efficient. Finally, registration is achieved by minimizing a variational chamfer-energy functional combined with the consistency regularizer. We demonstrate the effectiveness of our method on a number of experiments.

Keywords: meshless models, shape registration, shape correspondence, variational chamfer matching, distance transform

1. Introduction

Registering 2-D shapes that have been deformed by nonlinear mappings is a challenging problem that has applications in many areas including medical imaging [1] and shape recognition [2, 3]. Existing shape registration methods differ in three main components [4, 5]: shape representation, deformation model, and registration criterion. A number of recent works represent shape contours implicitly as zero level-sets of distance transforms [4, 5, 6, 7]. This shape representation has two important advantages. First, distance transform is a generic representation that can handle arbitrary shapes in arbitrary dimensions [5]. Secondly, distance transform provides a 2-D embedding of 1-D shape contours, so contours can be registered by aligning their 2-D distance transforms as done in image registration [4, 5, 6]. Here, the shape-registration criterion can be simply the sum-of-squared-differences (SSD) of the shapes' distance transforms [4] or mutual information (MI) [5]. Furthermore, nonlinear shape deformations can be represented using parametric models such as free-form deformation (FFD) [5] and radial basis functions (RBF) [1].

Despite these advances, using distance transforms for shape representation in registration methods present some common drawbacks. One issue is that distance transforms are redundant representations of shape contours which brings extra computational cost when performing registration. There is also the issue of stability of the representation when shapes undergo high-curvature deformation. Existing methods [4, 5, 6] use a narrow-band function to confine computation to be around the shape contours, but it only partially addresses the problem and complicates the registration framework. An additional issue is the use of regular mesh of control points to represent deformations. FFD-based deformation models [5, 6] rely on a mesh of regularly distributed control points that makes it difficult to adapt to the shape contours, and the registration accuracy may suffer from folding effect of the control-point mesh under large deformation. The limitation of control-point meshes is shared by many areas including computer graphics [8], engineering [9, 10], and image registration [11, 12]. The lack of flexibility of regular mesh models can be addressed by replacing the mesh-based deformation models with so-called meshless models that do not rely on explicit connections between control points. Examples of meshless models include thin-plate splines using radial basis functions (RBFs) that has long been used for image and shape registration [12, 1, 13, 14]. However, RBFs are less accurate than mesh models [10], and can be computationally expensive [12]. A number of recent works resorted to more accurate meshless models such as partition-of-unity (PU) method [10, 11, 9, 8]. The PU-meshless model represents a continuous function by blending together local polynomial models, and the local models can be arbitrarily distributed and overlaid without relying on connected control points [9]. However, existing PU methods rely on complicated functionals [11, 15] for regularizing shape deformation to avoid degenerating solutions and to produce smooth results.

In this paper, we follow the work in [4, 5, 6], and represent shape contours using distance transforms. In Section 3, we briefly introduce two representative registration methods [5, 4] using distance-transform representations, and elaborate on their limitations. Then, we address these limitations, and improve the state-of-the-art in two aspects. First, we improve the registration criterion, by proposing a variational extension to the classic chamfer-matching method [2] (Section 4). The proposed functional does not rely on a narrow-band function, and can better handle high-curvature shape deformations, leading to improvements in the registration accuracy. Chamfer-matching has been commonly used for detecting objects under affine transformations [2, 3, 16]. Here, we adopt it for variational nonrigid contour registration. In addition, we derive a simplified gradient function of the variational chamfer-matching functional to facilitate the registration process.

Secondly, we adopt the PU-meshless model to represent nonlinear shape deformations (Section 5). The meshless model can better represent large shape deformations, and can be easily adapted along shape contours to reduce computational cost. Figure 1 shows the overall scheme of our method. To regularize the PU model, we propose a novel regularizer that penalizes inconsistencies between neighboring local models. The inconsistency regularizer is much simpler than existing methods [11, 15], and provides a theoretical upper-bound to some regularizers [11]. It is worth pointing out that a similar consistency regularizer has been used for smoothing B-splines [17] that penalizes differences of adjacent B-splines' coefficients. However, this regularizer cannot be directly applied to PU-meshless model as it ignores the differences of local models' coefficients caused by shifting coordinate systems. Our method compensates for this shifting effect using a linear operator, so that neighboring polynomial models can be compared on a common ground.

In Section 6, we introduce our registration algorithm that combines the variational chamfer-matching gradients, the PU-meshless deformation model, and the consistency regularizer into a functional-minimization framework. We verify our method by comparing it to two recent registration meth-

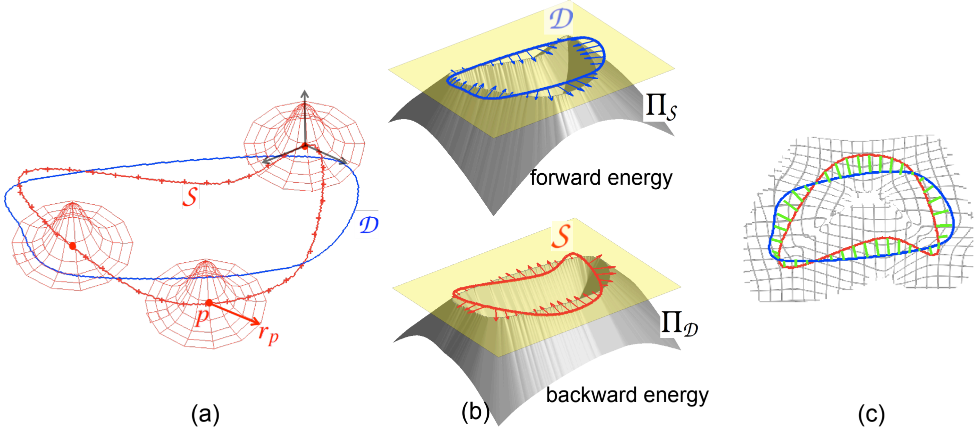


Figure 1: Meshless shape registration. (a) Source (red curve) and target (blue curve) shapes. Local polynomial deformation models are defined on overlapping patches along the contour. Patches are weighted by locally supported radial functions. Three patches are shown with their corresponding weighting functions. (b) Forward and backward registration gradients. (c) Blended global deformation map and correspondence after registration.

ods [5, 1]. The first method by Huang et al. [5] uses distance-transform representation and FFD-based deformation model, while the second method by Chen et al. [1] represents shapes using shape-context [13] and RBF-based deformation model. Experiments show that our method outperforms both methods in terms of registration accuracy. Additionally, our method can handle shapes with large high-curvature deformations (Section 7).

In summary, our main contribution is twofold. First, we improve the registration criterion used by previous shape-registration methods [4, 5, 6] that use distance-transform representation, and formulate the registration problem as the one of minimizing a variational chamfer-matching functional. Secondly, we introduce a meshless model for representing nonlinear shape deformations, and design a simple consistency regularizer to both produce smooth shape deformation and avoid degenerated results. Finally, we would like to comment that this work extends our previous works in [18, 19].

1.1. Our assumptions

We make an important assumption that shape contours are already "roughly aligned" before applying our registration algorithm. This is assumption is common to many existing shape and image registration methods that employ a global-to-local scheme where the shapes are registered in two stages [1, 5, 4]. In these methods, shapes are first roughly aligned using affine transformations to compensate for global shape deformations such as rotation, shifting, and scaling. Then, a second step seeks for local nonlinear deformations that align shapes as close as possible. Global shape alignment is not our main goal, since it can be achieved using off-the-shelf methods such as shape context [1, 13], mutual information [5], and chamfer matching [2]. As a result, we assume that shapes are aligned beforehand using a rigid transformation, and focus ourselves on the nonrigid registration part.

2. Related work

2.1. Shape matching and registration based on distance transform

The distance transform (distance map) is an implicit shape representation whose value at a point in map indicates the minimum distance of that point to the shape. This implicit representation has a number of advantages. First, the calculation of distance transforms is highly efficient [20]. Secondly, the distance information helps efficient shape alignment by allowing the searching step to be adjusted according to the shapes' mutual distance [2]. Indeed, chamfer-matching are frequently used for shape contour matching and detection [21, 3, 22, 16]. Earlier chamfer-matching methods [21, 3] focus on detecting shapes under affine transformations with little intra-class variations, while recent works include the use of orientation information [23, 16], hierarchical template matching [21], and statistical learning [24, 22, 25] that significantly improve the handling of shape variations and occlusion. The majority of these works apply distance transforms to shape contours, with exception of a recent work by Bai et al. [26] that uses distance transforms for matching articulated shape skeletons undergoing deformations.

Distance transforms are also used for registering contours [27, 4, 5, 6, 28, 29], and point sets [30]. Similarity can be drawn between the problems of shape registration and shape detection, since both problems involve shape matching, but important differences exist. The goal of shape detection is to locate a given shape in an image, while registration assumes the shape's existence and aims at accurately recovering its deformation. Paragios et al. [4] introduced the idea of using distance transform as an implicit embedding of shape contours, and registered distance maps essentially as normal 2-D images. Huang et al. [5] extended the method using FFD-based parametric deformation model. Munim et al. [7] extended the representation in [5] to a vectorized one for registering open shape contours. Rousson et al. [6] and Taron et al. [31] investigated the integration of statistical priors into the registration framework. In this paper, we focus on improving the registration criterion and parametric deformation model based on the work in [4, 5].

Finally, distance transform is a level-set representation that can naturally represent shapes of arbitrary topology [5]. One may find similarities between our work and level-set methods [32, 33] in three aspects. First, both methods use level sets of certain potential functions to represent a deforming contour. However, the potential functions in level-set methods are not necessarily distance maps, unless a regularizer is applied to the potential function to encourage it to maintain the shape of a distance transform [33]. In addition, many level-set methods aim at image segmentation (i.e., to locate a shape contour), and they deform a potential function according to an external force that favors image features (e.g., edges). In contrast, the shape registration methods in [4, 5, 6] and our method assume that two shape contours (i.e., the source and the target shapes) are given beforehand, and try to find *how* the source contour can be deformed into the target contour. Even though level-set methods are also used for object tracking [34, 35], their goal is to locate a moving object and seldom provide a dense correspondence between shape contours. Indeed, level-set methods can provide input data to our method by extracting the contours.

A second similarity between ours and level-set methods [32, 33, 34, 35] is the use of penalty functions for regularization. In level-set methods [32, 33], the regularizers penalize the geometric *properties* (e.g., curvature) of shape contours or their potential functions, to produce smooth shape boundaries. In contrast, registration methods [4, 5] penalize the *behavior* of shape contours to produce smooth shape deformation, and usually do not care about the shapes' curvature. Finally, both level-set methods and our work minimize variational functionals by iteratively solving partial differential equations (PDEs). Please refer to [4, 5] for more details of the relationship between level-set methods and shape registration based on distance transforms.

2.2. Modeling and regularizing nonlinear deformations

Distance transforms provide an implicit representation for shape contours themselves, but not of their deformations. It has been shown in [5] that FFD models can significantly improve registration accuracy over the nonparametric deformation model [4]. The FFD model belongs to a group of so-called mesh models [10] that rely on a mesh of explicitly connected control points, that makes it difficult to adapt mesh models to different shapes [36], especially for shapes undergoing topology changes, where remeshing is often required and can be prone to errors [37]. Another drawback of mesh models is that control points may collapse together (i.e., the folding effect) and change spatial relationship under large deformation, leading to numerical inaccuracies [11, 10]. For mesh models, the folding effect can be addressed by using extra regularizers to produce diffeomorphic deformations [38], but these regularizers increase the complexity of the registration framework.

To address these two issues, researchers in different areas seek for parametric models, called meshless models [10] that do not rely on explicit connections between control points. Meshless models can be easily adapted to shape contours and greatly alleviate the folding effect of large deformations. Please refer to [39] for an extensive review of meshless models. Here, we focus on the works that are more relevant to nonrigid registration. An example of meshless deformation models are the *Thin-Plate Splines* (TPS). These models represent a deformation field by a linear combination of radial basis functions [12] centered at scattered points, without explicit connections between them. However, it has been shown that RBFs can be numerically less accurate than the partition-of-unity method [10], due to the fact that RBFs cannot exactly represent a polynomial function (lack of reproducibility). In addition, it has been acknowledged in image registration [12] and computer graphics [37] RBFs have higher computational cost than the PU method.

In this paper, we adopt PU to represent nonrigid shape deformations. Additionally, we propose a simple regularizer to produce smooth deformation fields and to avoid degenerated results. Existing registration methods regularize the deformation model using a conformity functional [11] or by imposing mechanical energy constraints [40]. There are other meshless methods proposed for image registration [41, 42, 43, 44] and segmentation [45]. Many of these rely on predefined landmarks and use them as boundary conditions to solve mechanical PDEs. Finally, there are registration methods that represent shapes using PU models [46, 47]. These should not be confused with our method that use PU models to represent the shapes' deformations.

3. Distance functions and nonrigid registration

The goal of shape registration is to deform a source shape onto a target shape. This is achieved by searching for the best deformation field that minimizes a dissimilarity measure between the shapes. Formally, if $\mathcal{S}$ and $\mathcal{D}$ represent source and target shapes, respectively, and F is a dissimilarity measure between the two shapes, we seek for a warping field $\mathbf{u}(\mathbf{x})$ that satisfies the following equation:

$$\arg\min_{\mathbf{x}'} F(\mathcal{D}(\mathbf{x}), \mathcal{S}(\mathbf{x}'), \mathbf{x}'), \quad \mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is a coordinate vector. The dissimilarity measure F usually depends on the shape model. In this paper, we implicitly represent a shape $\mathcal{S}$ as the zero level set of its distance transform $\Pi_{\mathcal{S}}$ [4, 5], where $\mathcal{S}$ defines a partition of the image domain Ω. The model is given by:

$$\Pi_{\mathcal{S}} = \begin{cases} 0, & \mathbf{x} \in \mathcal{S} \\ +D_{\mathcal{S}}(\mathbf{x}) > 0, & \mathbf{x} \in R_S \\ -D_{\mathcal{S}}(\mathbf{x}) < 0, & \mathbf{x} \in [\Omega - R_S] \end{cases}, \tag{2}$$

where $D_{\mathcal{S}}$ is the minimum Euclidean distance between location $\mathbf{x}$ and shape $\mathcal{S}$, and R_S is the region inside $\mathcal{S}$. Distance transforms are essentially solutions of a Eikonal equation given by [20]:

$$\|\nabla \Pi_{\mathcal{S}}(\mathbf{x})\| = 1, \quad \mathbf{x} \in \mathbb{R}^2, \qquad \text{with } \Pi_{\mathcal{S}}(\mathbf{x}) = 0, \quad \mathbf{x} \in \mathcal{S}, \tag{3}$$

where shape contour $\mathcal{S}$ defines the boundary condition. Distance transforms are 2-D functions themselves, and rigorously $\Pi_{\mathcal{S}}(\mathbf{x})$ should be written as $\Pi_{\mathcal{S}(\mathbf{x})}(\mathbf{x})$, but to simplify the notation, we write as $\Pi_{\mathcal{S}}(\mathbf{x})$, $\Pi_{\mathcal{S}(\mathbf{x})}$ and even $\Pi_{\mathcal{S}}$. This simplification should not cause confusion given the context of its usage. The Eikonal equation in (3) is well-posed, meaning its solutions (distance transforms) depend continuously on the initial conditions (shape contours). As a result, similarity of distance transforms indicates the similarity of shape contours, and one can solve the shape registration problem indirectly by aligning shapes' distance transformations. Formally, the registration problem defined in (1) is converted to the following one given by [4]:

$$\arg\min_{\mathbf{x}'} F(\Pi_{\mathcal{D}}(\mathbf{x}), \Pi_{\mathcal{S}}(\mathbf{x}'), \mathbf{x}'), \quad \mathbf{x}' = \mathbf{x} + \mathbf{u}(\mathbf{x}). \tag{4}$$

In this approach, registering two 1-D contours is converted to the problem of registering their 2-D embeddings. The later problem can be easily solved using established 2-D image registration methods. In the following text, we will call this approach the *shape-embedding method*. Indeed, exactly as registering 2-D images, F can be simply defined as the squared-sum of differences, and registration is achieved by minimizing following functional [5, 4]:

$$E(\mathbf{u}) = \underbrace{\int_{\Omega} N_{\delta}(\Pi_{\mathcal{D}} - \Pi_{\mathcal{S}})^2 d\mathbf{x}}_{\text{data term}} + \alpha \underbrace{\int_{\Omega} N_{\delta} \left(\|\nabla u_x\|^2 + \|\nabla u_y\|^2 \right) d\mathbf{x}}_{\text{smoothness regularizer}} . \tag{5}$$

Here, $\Pi_{\mathcal{S}}$ and $\Pi_{\mathcal{D}}$ are distance transforms of the source and target shapes, respectively. u_x and u_y are the components of the deformation field, i.e., $\mathbf{u}(\mathbf{x}) = (u_x(\mathbf{x}), u_y(\mathbf{x}))$. First-order derivatives of u_x and u_y of large magnitude are penalized by a regularizer to produce a smooth deformation field and to avoid degenerated solutions. Finally, a proximity function N_{δ} limits the data-term evaluation to be near the shape's boundary. The use of a proximity function is a limitation of the shape-embedding methods, and the reason will be clarified in Section 3.1.

Similarly to nonrigid 2-D image registration, Equation 5 has been previously extended in several ways. First, the image deformation field $\mathbf{u}$ can be modeled parametrically using B-splines [5], or thin-plate splines [1]. Secondly, statistical priors can be leveraged to address uncertainties in the registration process [31]. Despite these developments, shape-embedding methods are still limited in some aspects. In the following section, we will elaborate on these limitations that motivated us for this work.

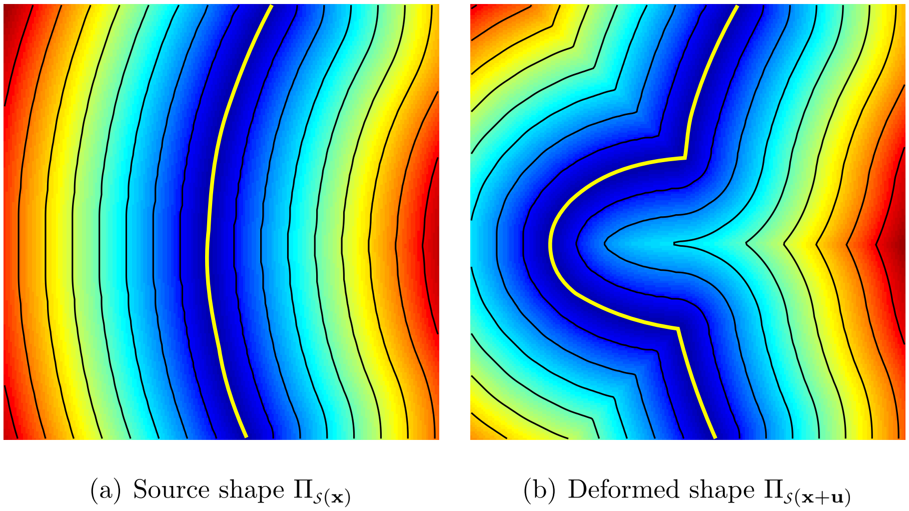
3.1. Limitations of the shape-embedding approach

The shape-embedding approach in Equation 5 facilitates the use of existing nonrigid 2-D registration techniques to solve 1-D shape registration problem. However, several issues need to be considered. First, as pointed out by El Munim and Farag [7], the distance map defined in Equation 2 cannot model open shape contours or shapes that do not define closed image domains (i.e., "inside" and "outside" parts of a shape). This problem was addressed in [7] using vector distance transform. However, for the purpose of contour registration, we may simply use an unsigned distance transform. In this paper, we assume the shapes to be 0-1 encoded contours, and represented using unsigned distance transform as:

$$\Pi_{\mathcal{S}} = \begin{cases} 0, & \mathbf{x} \in \mathcal{S} \\ D_{\mathcal{S}}(\mathbf{x}) > 0, & \mathbf{x} \in \Omega - \mathcal{S} \end{cases}, \tag{6}$$

where $D_{\mathcal{S}}(\mathbf{x})$ is still the minimum Euclidean distance of position $\mathbf{x}$ to the shape contour $\mathcal{S}$. In the following text, we always assume $\Pi_{\mathcal{S}} \geq 0$.

Secondly, minimizing Equation 5 essentially deforms the distance map $\Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u})$ to align with $\Pi_{\mathcal{D}}(\mathbf{x})$. However, deformation in the distance transform only approximates the shape deformation. Formally, $\Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u}) \neq \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$, where $\Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u})$ is a direct deformation of the original shape's distance map, while $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ is the distance map calculated from the deformed shape. For example, it has been noticed by Paragios et al. [4] that scaling a shape is not equivalent to scaling its distance transform. Here, we would like to further point out that modest deformations in the shape contour (i.e., $\mathcal{S}(\mathbf{x}+\mathbf{u})$) may sometimes cause significant deformations in its distance transform (i.e., $\Pi_{\mathcal{S}(\mathbf{x}+u)}$), since $\mathcal{S}(\mathbf{x}+\mathbf{u})$ is the boundary condition for the Eikonal PDE (Equation 3), its deformation will be propagated into the PDE's solution. As a result, registering distance maps may be unnecessarily complicated. Figure 2 shows the distance maps of a bending curve, i.e., $\Pi_{\mathcal{S}(\mathbf{x})}$ and



(a) Source shape $\Pi_{\mathcal{S}(\mathbf{x})}$

(b) Deformed shape $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$

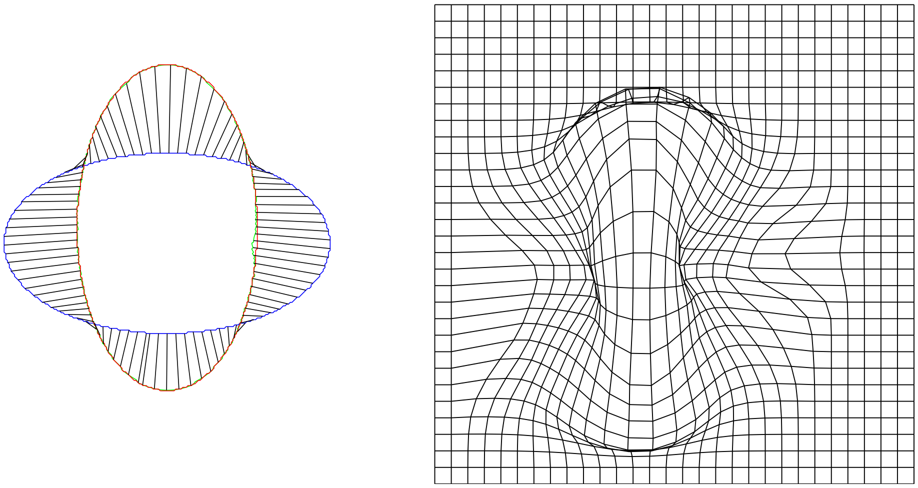
Figure 2: The (unsigned) distance maps of a bending curve (center in yellow). The distance maps are color-coded, with red and blue indicating maximum and minimum values, respectively. Black iso-curves are plotted to indicate points of the same distance values. (a) Distance map of the source shape. (b) A modest bending in the shape contour may cause high-curvature distortions in the distance map.

$\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$, together with iso-curves indicating points of same distance values. The deformation of $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ has much higher curvature in certain places than the deformation of $\mathcal{S}(\mathbf{x}+\mathbf{u})$ itself. As a result, when directly registering the distance maps, much higher bending energy is needed to cope with these high curvature regions, leading to registration errors or even degenerated results.

This problem is only partially addressed in [4, 5] by using a proximity function N_δ (Equation 5), that limits similarity measure in the proximity of shape contours with distance value less than δ, where the propagated distortion is minimal. Intuitively, N_δ defines a “narrow band” along the shape contour to exclude the high-curvature part. Nevertheless, care must be taken to choose the narrow-band width parameter δ. On one hand, if δ is too small, there may be little overlapping areas between the two shapes, and the registration algorithm can easily get trapped in local minima. On the other hand, if δ is too large, the proximity function may fail to exclude distorted regions. Taron et al. [31] proposed to use an iteratively decreasing δ, but this approach complicates the optimization algorithm and still relies on this extra parameter.

Finally, registering 2-D distance functions results into extra computation as the method needs to register the whole image plane instead of 1-D shape contours. Parametric representations [5] such as B-Splines can be used to improve registration efficiency, but B-Spline model cannot be easily adapted to the shape contours as it relies on a regular grid (mesh) of control points. Consequently, a lot of computation is still wasted in modeling and calculating deformation fields at irrelevant regions. The use of a proximity function [4, 5, 31] further reduces the computational cost, but the formulation becomes more complicated. More importantly, mesh-based models may suffer from the folding effect when the deformation is large. Figure 3 shows a result obtained using the method in [5], with the deformation field represented using a control-point mesh of B-splines [1]. Large deformations in parts of the image cause some control points to be "squeezed" together and even flip their positions (folding). As a result, the deformation field no longer equals to the interpolation of control-point displacements, leading to less accurate

[1] Obviously, the shape deformation here can be modeled as affine global deformation, but we treat it as large local deformation for the sake of demonstration.



(a) Correspondence

(b) Deformation field

Figure 3: Folding of mesh model under large deformation. The images are obtained using method in [5]. a) Correspondence of the source shape (in blue) deformed to the target image (in red). The deformed source shape is in green. b) Under large deformation, the B-spline control points collapse together and may even flip their positions.

or even failed registration.

Existing registration works address the folding issue in two ways. First, foldings in mesh models can be removed with a stronger regularizer to produce diffeomorphic deformation fields that are not only smooth but also invertible [38]. For example, the folding areas shown in Figure 2(b) violate one-to-one correspondence between the original and target images (thus not invertible), and will be penalized by a diffeomorphic regularizer. The main drawback of diffeomorphic registration methods is that they are relatively more difficult to optimize.

Alternatively, the folding problem can also be alleviated by replacing the mesh models with so-called “meshless models” that do not rely on explicit connections between control points, and thus are less sensitive to the positioning of control points. Meshless models were previously used in mechanical engineering [9, 10] for solving partial differential equations (PDEs), and have been recently introduced in computer graphics [8] and image registration [11]. It has been shown that meshless models can improve computational accuracy under large deformations [10, 11]. In this paper, we extend a meshless model called partition-of-unity method [9, 8, 11] and adopt it for shape registration.

In summary, the registration methods in [4, 5, 31] share two common drawbacks: the reliance on a narrow-band function that complicates the registration framework, and the use of a mesh-based deformation model with limited adaptation to the shape contour, in addition to causing a folding effect under large deformations. Next, we propose a novel dissimilarity measure based on variational formulation of the classic chamfer-matching energy that does not rely on a proximity function, and in Section 5, we integrate this energy term with a meshless parametric-deformation model that can be naturally adapted to the shape contours, and also reduces the folding effect.

4. Variational chamfer-matching energy

The limitations of shape-embedding methods are partially due to the fact that their distance transforms are registered as normal 2-D images. In this case, the dissimilarity measure takes into consideration the entire embedding 2-D space, leading to a redundancy that unnecessarily distorts the contour-matching quality. The narrow-band function only partially reduces this problem. To completely remove this redundancy, we need a dissimilarity function that directly measures the shape alignment "along" the source and target contours without taking into consideration areas around the shapes. Meanwhile, we still want to use distance transform to represent the shapes due to its many advantages.

We start by observing that, when the source shape $\mathcal{S}$ is aligned with the target $\mathcal{D}$, the deformed shape $\mathcal{S}(\mathbf{x}+\mathbf{u})$ will coincide with the zero level set of $\Pi_{\mathcal{D}}$, i.e., $\mathcal{S}(\mathbf{x}+\mathbf{u})\ \Pi_{\mathcal{D}}(\mathbf{x}) = 0$. Consequently, we may enforce alignment between shapes by minimizing the squared sum $\int_{\Omega} |\mathcal{S}(\mathbf{x}+\mathbf{u})\ \Pi_{\mathcal{D}}|^2\, d\mathbf{x}$, which corresponds to the classic chamfer-matching energy function [2], used for affine shape registration. However, this functional can be ill-posed. For example, the energy function will vanish for any deformation field that shrinks the source shape to a single point on shape contour $\mathcal{D}$.

We can address this problem by including a symmetric term that measures the distance-error between target and source shapes, similarly to the classic *symmetric chamfer-matching energy* [3]. In addition, we compensate for scaling by dividing the distance-error by the contours' lengths, and minimize the following functional given by:

$$\begin{aligned} E^{\mathrm{d}}(\mathbf{u}) &= \frac{1}{A_{\mathcal{S}}}\int_{\Omega} |\mathcal{S}(\mathbf{x}+\mathbf{u})\ \Pi_{\mathcal{D}}|^2 d\mathbf{x} + \frac{1}{A_{\mathcal{D}}}\int_{\Omega} |\mathcal{D}(\mathbf{x})\ \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}|^2 d\mathbf{x} \\ &= \underbrace{\frac{1}{A_{\mathcal{S}}}\int_{\Omega} \mathcal{S}(\mathbf{x}+\mathbf{u})\ \Pi^2_{\mathcal{D}} d\mathbf{x}}_{\text{forward energy } E^{\mathrm{f}}} + \underbrace{\frac{1}{A_{\mathcal{D}}}\int_{\Omega} \mathcal{D}(\mathbf{x})\ \Pi^2_{\mathcal{S}(\mathbf{x}+\mathbf{u})} d\mathbf{x}}_{\text{backward energy } E^{\mathrm{b}}}, \end{aligned} \tag{7}$$

where $A_{\mathcal{S}} = \int_{\Omega} \mathcal{S}(\mathbf{x}+\mathbf{u}) d\mathbf{x}$ and $A_{\mathcal{D}} = \int_{\Omega} \mathcal{D}(\mathbf{x}) d\mathbf{x}$ are contour lengths as normalizing factors, and the second equation in (7) holds because $\mathcal{S}$ is a binary map and $\mathcal{S}(\mathbf{x})^2 = \mathcal{S}(\mathbf{x})$. Additionally, $E^{\mathrm{d}}(\mathbf{u})$ is independent of the sign of Π, and we have assumed that $\Pi_{\mathcal{S}} \geq 0$ and $\Pi_{\mathcal{D}} \geq 0$ (unsigned distance transform). The registration error is now directly measured using the shape contours without resorting to a proximity function as in [4, 5, 31]. As in the classic chamfer matching, the use of distance transform allows for efficient optimization [4, 5] by providing a distance-adaptive gradient function. Figure 1(b) shows the two energy terms in our shape-registration functional. We now examine the minimization of our variational chamfer-matching energy.

4.1. Functional minimization using Euler-Lagrange equation

A common approach to minimizing a variational functional is to calculate its Euler-Lagrange equation [48]. For example, if the functional is given by:

$$E(\mathbf{u}) = \int L\left(\mathbf{x}, \mathbf{u}(\mathbf{x}), \dot{\mathbf{u}}(\mathbf{x})\right) d\mathbf{x}, \tag{8}$$

then its Euler-Lagrange equation is calculated as follows:

$$\mathbf{J} = \frac{\partial L}{\partial \mathbf{u}} - \frac{d}{d\mathbf{x}} \left(\frac{\partial L}{\partial \dot{\mathbf{u}}(\mathbf{x})} \right) = 0. \tag{9}$$

The Euler-Lagrange equation can rarely be solved directly, and it is often solved iteratively by converting the original problem to the equivalent PDE:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{J}(\mathbf{u}). \tag{10}$$

In solving Equation 10, $\mathbf{J}$ is used as a gradient function to facilitate iterative gradient-descent methods. However, calculating the gradients $\mathbf{J}$ from the variational functional in Equation 7 is not always straightforward. In these cases [32], we need to look for equivalent or approximating constraints to its exact Euler-Lagrange equation. In next section, we examine the forward and backward energy terms separately, and derive a simple gradient function for minimizing Equation 7. We begin with an intuitive and informal derivation, and then provide a more rigorous prove in Section 4.3 that the derived function can indeed minimize the variational chamfer-matching energy.

4.2. Simplified chamfer-matching gradient

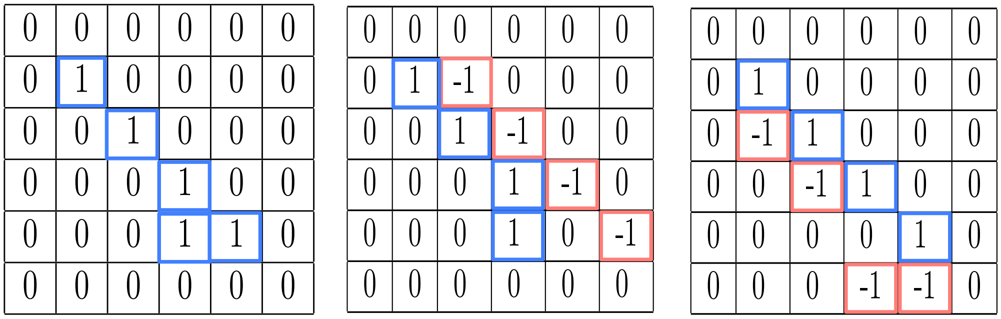
Starting with our forward energy $E^f(\mathbf{u})$, its Euler-Lagrange equation can be written as follows:

$$\frac{\partial L^{\mathrm{f}}}{\partial \mathbf{u}} = \Pi^2_{\mathcal{D}(\mathbf{x})} \frac{\partial \mathcal{S}(\mathbf{x}+\mathbf{u})}{\partial \mathbf{x}} = 0 \tag{11}$$

where $\mathcal{S}$ is the source shape contour represented as a binary edge map. However, $\frac{\partial \mathcal{S}(\mathbf{x}+\mathbf{u})}{\partial \mathbf{x}}$ does not exist in the classic limiting sense and is numerically sensitive, since edge maps $\mathcal{S}$ and $\mathcal{D}$ are not even continuous as 2-D functions, let alone differentiable. This is the price we pay for our varitationl chamfer-matching formulation. Fortunately, under integration, the derivative of $\mathcal{S}(\mathbf{x}+\mathbf{u})$ can be exchanged to that of $\Pi^2_{\mathcal{D}}$, according to the generalized derivative based on distribution theory [49] as:

$$\begin{aligned} \int \frac{\partial L^{\mathrm{f}}}{\partial \mathbf{u}}\, d\mathbf{x} &= \int \Pi^2_{\mathcal{D}}(\mathbf{x}) \frac{\partial \mathcal{S}(\mathbf{x}+\mathbf{u})}{\partial \mathbf{x}}\, d\mathbf{x} \\ &= \int -\frac{\partial \Pi^2_{\mathcal{D}}(\mathbf{x})}{\partial \mathbf{x}} \mathcal{S}(\mathbf{x}+\mathbf{u})\, d\mathbf{x} \\ &= \int -2\Pi_{\mathcal{D}}(\mathbf{x}) \frac{\partial \Pi_{\mathcal{D}}(\mathbf{x})}{\partial \mathbf{x}} \mathcal{S}(\mathbf{x}+\mathbf{u})\, d\mathbf{x}. \end{aligned} \tag{12}$$

The rationale behind this conversion is illustrated in Figure 4 by a simple numerical example. Figure 4(a) shows a 0–1 encoded binary map of a shape contour, with its finite differences along x and y directions shown in Figure 4(b) and Figure 4(c), respectively. It is intuitive that these difference maps reproduce the negative derivative operator along the shape contour.



0	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	0	0	0	0

(a) $\mathcal{S}(\mathbf{x})$

0	0	0	0	0	0
0	1	-1	0	0	0
0	0	1	-1	0	0
0	0	0	1	-1	0
0	0	0	1	0	-1
0	0	0	0	0	0

(b) $\partial\mathcal{S}(\mathbf{x})/\partial x$

0	0	0	0	0	0
0	1	0	0	0	0
0	-1	1	0	0	0
0	0	-1	1	0	0
0	0	0	0	1	0
0	0	0	-1	-1	0

(c) $\partial\mathcal{S}(\mathbf{x})/\partial y$

Figure 4: Gradients of a 0–1 edge map reproduce the negative derivative operators. When integrated with distance map $\Pi_{\mathcal{D}}$, the operators are essentially transfered to $\Pi_{\mathcal{D}}$.

After integration with the target shape's distance map $\Pi_{\mathcal{D}}(\mathbf{x})$, the derivative operators are essentially transfered over to $\Pi_{\mathcal{D}}(\mathbf{x})$. In this way, we may replace the original gradient operator with a "better-behaved" one on $\Pi_{\mathcal{D}(\mathbf{x})}$, since $\Pi_{\mathcal{D}(\mathbf{x})}$ is differentiable. Equation 12 suggests that we may approximate the original Euler-Lagrange equation in (11) by a simpler constraint as:

$$-2\Pi_{\mathcal{D}}\frac{\partial\Pi_{\mathcal{D}}}{\partial\mathbf{x}}\mathcal{S}(\mathbf{x}+\mathbf{u}) = 0. \tag{13}$$

Let us now examine the Equation 13 in detail. From the definition of distance transform, we know that $\left\|\frac{\partial\Pi_{\mathcal{D}(\mathbf{x})}}{\partial\mathbf{x}}\right\| = 1$, i.e., $\frac{\partial\Pi_{\mathcal{D}(\mathbf{x})}}{\partial\mathbf{x}}$ is a *direction vector* that points to the increasing value of $\Pi_{\mathcal{D}(\mathbf{x})}$. Taking the negative sign into consideration, the left-hand side of Equation 13 is a vector pointing in the direction that minimizes the distance of $\mathcal{S}(\mathbf{x}+\mathbf{u})$ to $\mathcal{D}(\mathbf{x})$, and the magnitude of this vector varies along the shape contour, proportionally to the distance of $\mathcal{S}(\mathbf{x}+\mathbf{u})$ towards the target shape, i.e., $\Pi_{\mathcal{D}(\mathbf{x})}\mathcal{S}(\mathbf{x}+\mathbf{u})$. Figure 5(b) shows the distribution of these vectors. They resemble a force that "pulls" the source shape to align with the target, hence we call it the *forward force*.

We now turn our attention to the backward-energy term E^{b}. From the definition of E^{b} in (7), we obtain:

$$\frac{\partial L^{\mathrm{b}}}{\partial \mathbf{u}} = 2\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} \frac{\partial \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{u}} \mathcal{D}(\mathbf{x}) = 0, \tag{14}$$

where the distance map $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ is a solution of the Eikonal equation in (3), given $\mathcal{S}(\mathbf{x}+\mathbf{u})$ as its boundary condition, so the dependence of $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ on variations in $\mathbf{u}$ (i.e., $\frac{\partial \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{u}}$) is difficult to analyze. This dependence can be greatly simplified if we assume that the deformation field is a simple shifting, i.e., $\mathbf{u}(\mathbf{x}) = \mathbf{u}_0$, $\forall \mathbf{x} \in \Omega$ being a constant function. With this assumption, the shape deformation equals to the deformation of its distance map [4]: $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} = \Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u})$. Substituting this into (14), we obtain:

$$\begin{aligned}\frac{\partial L^{\mathrm{b}}}{\partial \mathbf{u}} &= 2\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} \frac{\partial \Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u})}{\partial \mathbf{u}} \mathcal{D}(\mathbf{x}) \\ &= 2\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} \frac{\partial \Pi_{\mathcal{S}}(\mathbf{x}+\mathbf{u})}{\partial \mathbf{x}} \mathcal{D}(\mathbf{x}) \\ &= 2\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} \frac{\partial \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{x}} \mathcal{D}(\mathbf{x}).\end{aligned} \tag{15}$$

We would like to clarify that $\frac{\partial \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{x}}$ in (15) is simply the gradient of $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ without further dependence on the deformed shape $\mathcal{S}(\mathbf{x}+\mathbf{u})$. Using a more complex notation, this gradient term should be written as $\frac{\partial \Pi_{\mathcal{S}'}(\mathbf{x})}{\partial \mathbf{x}}$ where $\mathcal{S}' = \mathcal{S}(\mathbf{y}+\mathbf{u})$ is the deformed shape. Here, we keep its notation simpler and consistent with Equation 13. It is also worth pointing out that the above derivation is valid only under the assumption that $\mathbf{u}(\mathbf{x})$ is constant for every $\mathbf{x}$, but we relax this assumption to arbitrary continuous $\mathbf{u}(\mathbf{x})$, and use (15) as the gradient function for minimizing the backward energy. This is of course only an approximated function as we have ignored variations of the deformation function $\mathbf{u}(\mathbf{x})$, but we show in the following section that this simplification is not arbitrary and the approximated gradient function will lead to the minimization of our chamfer-matching functional.

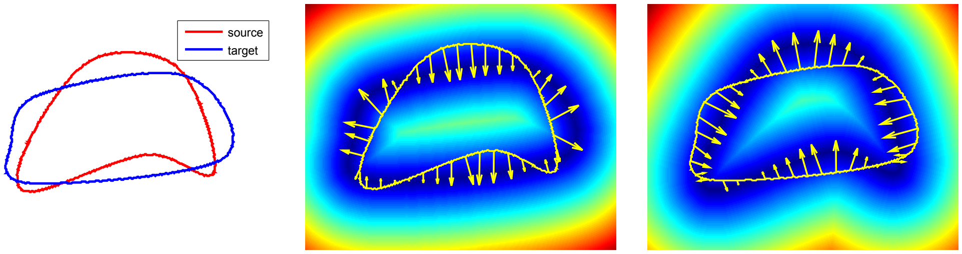
The gradient function in Equation 15 has similar intuitive motivation as the forward force. By negating the left-hand side of Equation 15, we obtain $-\frac{\partial L^{\mathrm{b}}}{\partial \mathbf{u}}$ that acts like a force "pulling" the target shape towards the source as shown in Figure 5(c). In actual minimization, the target shape is kept unchanged (static), thus the force is applied in reversed direction to the source shape, and we call it the *backward force*.

Combining (13) with (15) and ignoring the constants, we may now approximate the Euler-Lagrange equation of the original variational functional (Equation 7) as follows:

$$\mathbf{J} \approx \widetilde{\mathbf{J}} = \underbrace{-\Pi_{\mathcal{D}(\mathbf{x})} \frac{\partial \Pi_{\mathcal{D}(\mathbf{x})}}{\partial \mathbf{x}} \mathcal{S}(\mathbf{x}+\mathbf{u})}_{\text{forward force}} + \underbrace{\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})} \frac{\partial \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}}{\partial \mathbf{x}} \mathcal{D}(\mathbf{x})}_{\text{backward force}} . \tag{16}$$

$\widetilde{\mathbf{J}}$ closely resemble the gradient of classic chamfer-matching method [2], and we call $\widetilde{\mathbf{J}}$ the *variational chamfer-matching gradient*. $\widetilde{\mathbf{J}}$ can also be interpretated as a force field deforming the source contour, in a similar analogy used in the deformation of active contours [32]. Finally, the strength of the combined force increases according to their mutual distances.

Note that, in the chamfer-matching energy functional (Equation 7), we may also use the $\mathbf{L}^1$ norm instead of the squared-sum (i.e., $\mathbf{L}^2$ norm). In this case, the forward and backward forces degenerate to direction vectors with uniform magnitude regardless to the mutual distance between the shapes. Our experiments showed that $\mathbf{L}^1$ norm is more sensitive to local minima,



(a) (b) (c)

Figure 5: Variational chamfer-matching gradient. a) Source and target shape contours. b) Forward force is distributed along the source shape (curve in yellow). The distance transforms are color-coded, with red and blue indicating the maximum and minimum values respectively. The forward force pulls the source towards the target. c) Backward force in reversed direction that pulls the target to the source. The magnitude of these forces are proportional to the distances between two shape contours.

and leads to slower minimization convergence. This observation echoes a similar finding in the classic chamfer-matching method [2].

4.3. Variational chamfer-matching based on the simplified gradient

We have derived an approximate but simple gradient function $\widetilde{\mathbf{J}}$ in (16) to minimize the variational chamfer-matching functional $E^{\mathrm{d}}(\mathbf{u})$. Now, we show that $\widetilde{\mathbf{J}}$ can indeed lead to the minimization of $E^{\mathrm{d}}(\mathbf{u})$. From variational calculus, we know that a necessary condition for minimizing $E^{\mathrm{d}}(\mathbf{u})$ is its Euler-Lagrange equation $\mathbf{J} = 0$. However, this condition is not sufficient as the Euler-Lagrange equation may also hold when $E^{\mathrm{d}}(\mathbf{u})$ is at local minima. Nevertheless, the Euler-Lagrange equation is widely used in computer vision to minimize variational functionals. Here, we wish to establish a similar relationship between the minimization of $E^{\mathrm{d}}(\mathbf{u})$ and our approximate gradent function $\widetilde{\mathbf{J}}$, then the use of $\widetilde{\mathbf{J}}$ is theoretically justifiable. It turns out that $\widetilde{\mathbf{J}} = 0$ is indeed a necessary and sufficient condition for the variational-matching energy to vanish, i.e., $E^{\mathrm{d}}(\mathbf{u}) = 0$.

Before elaborating on the proof, we want to clarify that our discussion is limited to continuous deformation fields, i.e., $\mathbf{u} \in C(\Omega)$ with $C(\Omega)$ as the set of continuous functions defined on domain Ω. Vanishing of the chamfer-matching energy, i.e., $\min\limits_{\mathbf{u}\in C(\Omega)} E^{\mathrm{d}}(\mathbf{u}) = 0$ can only be reached when two shapes $\mathcal{S}(\mathbf{x}+\mathbf{u})$ and $\mathcal{D}(\mathbf{x})$ can be exactly aligned, meaning that there exists a *continuous* deformation $\mathbf{u}(\mathbf{x})$, such that $\mathcal{S}(\mathbf{x}+\mathbf{u}) = \mathcal{D}(\mathbf{x})$, $\forall \mathbf{x} \in \Omega$ [2]. A mismatch of the two shapes at any image pixel $\mathbf{p} \in \Omega$ would lead to $E^{\mathrm{d}}(\mathbf{u}) > 0$, due to the non-negativeness of unsigned distance transform. As a result, if the source and target shapes have different topologies, then $E^{\mathrm{d}}(\mathbf{u}) > 0$, $\forall \mathbf{u}(\mathbf{x}) \in C(\Omega)$ since there is no continuous mapping between shapes of different topologies. By focusing on $E^{\mathrm{d}}(\mathbf{u}) = 0$, we have excluded the cases of aligning shapes with different topologies. Fortunately, these cases are not very interesting for many practical applications. When the two shapes have the same topology, then $E^{\mathrm{d}}(\mathbf{u}) = 0$ is indeed the global minimum of $E^{\mathrm{d}}(\mathbf{u})$. In summary, we can prove the following lemma:

Lemma 4.1. *If* $\min\limits_{\mathbf{u}\in C(\Omega)} E^{d}(\mathbf{u}) = 0$*, then* $E^{d}(\mathbf{u}) = 0 \Leftrightarrow \widetilde{\mathbf{J}}(\mathbf{u}) = 0$.

Proof On one hand, $E^{\mathrm{d}}(\mathbf{u}) = 0$ means that a continuous deformation field $\mathbf{u}(\mathbf{x})$ exists such that $\mathcal{S}(\mathbf{x}+\mathbf{u}) = \mathcal{D}(\mathbf{x})$, and $\Pi_{\mathcal{D}(\mathbf{x})} = \Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$. By the definition of distance transform, we obtain $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}\, \mathcal{D}(\mathbf{x}) = 0$, $\Pi_{\mathcal{D}(\mathbf{x})}\, \mathcal{S}(\mathbf{x}+\mathbf{u}) = 0$ and

[2] Rigorously, exceptions may exist for singular point set of zero measure. As far as discrete image data is concerned, this assertion is valid.

$\widetilde{\mathbf{J}}(\mathbf{x}) = 0$. Consequently, $\widetilde{\mathbf{J}}(\mathbf{x}) = 0$ is indeed necessary for $E^{\mathrm{d}}(\mathbf{u}) = 0$.

On the other hand, let us assume $\widetilde{\mathbf{J}}(\mathbf{x}) = 0$, $\forall \mathbf{x} \in \Omega$, then we can show that $\mathcal{S}(\mathbf{x}+\mathbf{u}) = \mathcal{D}(\mathbf{x})$ must be true. For example, if $\mathcal{S}(\mathbf{x}+\mathbf{u}) \neq \mathcal{D}(\mathbf{x})$ for some $\mathbf{x} \in \Omega$, we can assume $\mathcal{D}(\mathbf{x}) = 1$ and $\mathcal{S}(\mathbf{x}+\mathbf{u}) = 0$ without loss of generality. As a result, the forward force vanishes at $\mathbf{x}$, from the definition of $\widetilde{\mathbf{J}}(\mathbf{x})$ (Equation 16). Since $\widetilde{\mathbf{J}}(\mathbf{x}) = 0$, the backward force also become zero at $\mathbf{x}$, i.e., $-\Pi_{\mathcal{D}(\mathbf{x})}\frac{\partial \Pi_{\mathcal{D}(\mathbf{x})}}{\partial \mathbf{x}} = 0$, but we have assumed that $\mathcal{D}(\mathbf{x}) = 1$, so $\Pi_{\mathcal{D}}(\mathbf{x}) > 0$ and $\left|\frac{\partial \Pi_{\mathcal{D}(\mathbf{x})}}{\partial \mathbf{x}}\right| = 1$, leading to a contradiction. Consequently, $\widetilde{\mathbf{J}}(\mathbf{x}) = 0$ ensures that $\mathcal{S}(\mathbf{x}+\mathbf{u}) = \mathcal{D}(\mathbf{x})$ holds for all $\mathbf{x} \in \Omega$. This in turn proves $E^{\mathrm{d}}(\mathbf{u}) = 0$, and hence the sufficiency of $\widetilde{\mathbf{J}} = 0$ for $E^{\mathrm{d}}(\mathbf{u}) = 0$ $\square$.

We have shown that $\widetilde{\mathbf{J}}(\mathbf{u})$ can be used to minimize the variational chamfer-matching energy. Now, we can combine the chamfer-matching energy with different regularizers and deformation models for nonrigid shape registration such as the derivative-based regularizer of Equation 5 and B-Spline representation in [5]. However, B-Spline models rely on an explicitly connected control-point grid (i.e., mesh), making it difficult to adapt the model to shapes of challenging topologies (holes, splits). Instead, the whole image domain is taken into consideration even though a large part of it can be irrelevant. Additionally, B-Spline models may suffer from the folding effect that control points collapse together under large deformation, causing numerical instability. Next, we study how to remove the redundancy from parametric deformation models, by adopting a meshless representation that approximates the shape's deformation field.

5. Meshless deformation model

Our model is derived from recent developments in computer graphics [8] and mechanical engineering [10], on building shape functions of arbitrary topology. Specifically, our model belongs to a group of so-called *partition-of-unity* meshless methods. Although there are other meshless shape-deformation models based on thin-plate splines and radial basis functions (RBFs) [1], those methods are less accurate than polynomial-based representations as radial basis functions cannot exactly represent polynomial deformations (lack of reproducibility) [10]. In the partition-of-unity model, one can ensure polynomial reproducibility by locally modeling shape deformation as polynomials. These local deformation models are then blended together into a global deformation field based on a set of weighting functions. We illustrate the partition-of-unity blending concept using the 1-D example shown in Figure 5. The figure shows 1-D patches represented by their weighting functions partitioning the data domain into three regions. The local deformation models are represented by the colors pink, blue, and red. The global deformation model is represented by the green polynomial curve obtained by blending the local models. Here, we would like to remind the readers that the meshless model in our method is used for representing the incremental nonrigid deformation (i.e., function $\mathbf{u}$ in Equation 7), but not the shape contours themselves (the shapes are represented by distance functions and binary maps). Our approach largely follows the partition-of-unity used in computer graphics [8]. In the following subsections, we first introduce the local deformation model, and then explain how to blend them into a global model.

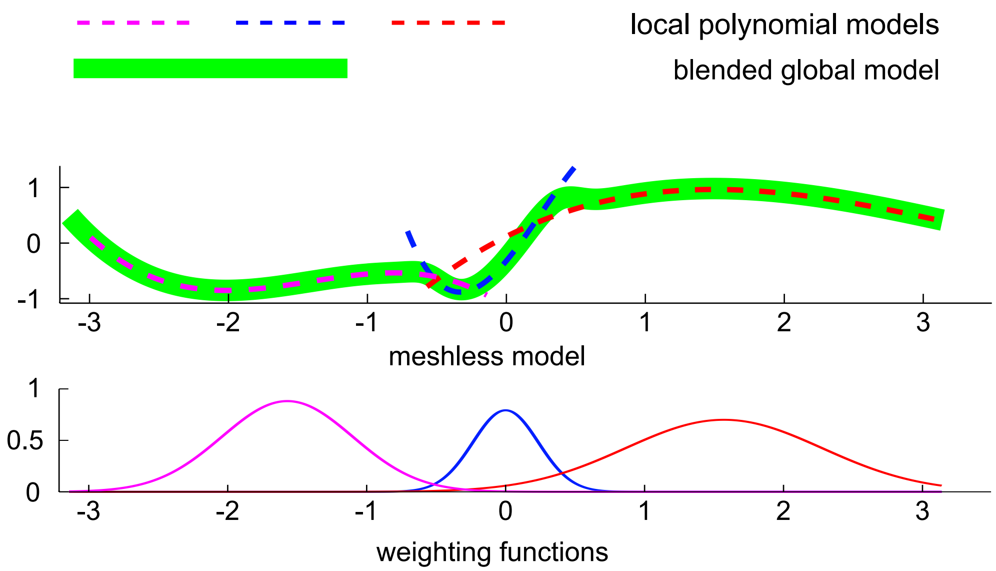


Figure 6: A 1-D meshless model based on partition-of-unity. The local polynomial models are blended together (top), using predefined weighting functions (bottom).

5.1. Local polynomial model

We commence by modeling shape deformation locally in scattered local domains called patches [9]. As in [8], we define these patches as the support of some weighting functions. Various types of weighting functions exist [10]. In this work, we use the 2nd-order B-Spline $w_p(\mathbf{x}) = \alpha_p \, w_b \left(\frac{\|\mathbf{x}-\mathbf{p}\|}{r_p} \right)$ for its computational efficiency, where $\alpha_p \in (0, 1]$ is the patch's predefined influence factor in the final global blending. Formally, a 2nd-order B-Spline function is defined as a piecewise polynomial:

$$w(r) = \begin{cases} \frac{3}{4} - r^2 & , r < \frac{1}{2} \\ \frac{1}{2}(\frac{3}{2} - r)^2 & , \frac{3}{2} \geq r \geq \frac{1}{2} \\ 0 & , r > \frac{3}{2} \end{cases}, \tag{17}$$

This weighting function's support is a disk with radius r_p centered at $\mathbf{p}$ (ball in 3-D[3]). To summarize, a disk-like patch p is defined by three parameters $(\mathbf{p}, r_p, \alpha_p)$, i.e., its center position, radius, and influence factor. The union of these patches covers the image region of interest, Ω, i.e., $\Omega \subset \cup_p \text{supp}\,(w_p)$, $p \in \mathbb{P}$, with $\mathbb{P}$ is the set of all patches. Note that although the weighting function in (17) is a radial function, its usage is different from previous RBF models such as thin-plate splines [1]. Here, RBFs are used for blending local polynomial models instead of directly representing the shape deformation.

It is also important to point out that the patches simply define a partition of the computational domain, and remain static during shape deformation, unlike B-spline control points, whose displacements directly control the deformation field. Instead, the local deformation $\mathbf{u}_p = (u, v)$ within each patch is defined by the parameters of a polynomial model centered at $\mathbf{p}$ as follows:

$$u(\mathbf{x}) = \sum_{s,t=0}^{s,t=m} a_{s,t} x^s y^t \qquad \text{and} \qquad v(\mathbf{x}) = \sum_{s,t=0}^{s,t=m} b_{s,t} x^s y^t, \tag{18}$$

Alternatively, $\mathbf{u}_p(\mathbf{x})$ is a linear combination of monomial basis functions $\boldsymbol{\phi}^\mathsf{T}(\mathbf{x}) = (1, x, y, xy,\, x^2, y^2, \ldots, x^m y^m)$, with a two-column coefficient matrix

[3] A 3-D extension is straightforward.

$\mathbf{d}_p$ as:

$$\mathbf{u}_p(\mathbf{x}) = {\mathbf{d}_p}^\mathsf{T}\boldsymbol{\phi}_p(\mathbf{x}), \tag{19}$$

where $\operatorname{vec}(\mathbf{d}_p) = (a_{0,0}, b_{0,0}, \cdots, a_{m,m}, b_{m,m})^\mathsf{T}$ and $\boldsymbol{\phi}_p(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x} - \mathbf{p})$ is the basis vector re-centered at $\mathbf{p}$. The sequence of monomials in $\boldsymbol{\phi}$ can be represented as a one-to-one mapping from the natural number to nonnegative 2-D monomial indices $\xi : \mathbb{N} \leftrightarrow \mathbb{N}_0 \times \mathbb{N}_0$. In this paper, the monomials are arranged in a Pascal-triangle manner [10], where low-order monomials are arranged before high-order ones. For monomials of the same order, the ones with more balanced indices in x and y are arranged first. This special arrangement increases numerical stability according to existing literature [10].

5.2. Blending local models into a global deformation field

Once the local deformation models are at hand, the deformation at any point $\mathbf{x}$ is obtained by blending local fields of patches that contain $\mathbf{x}$. These patches are denoted by $N_\mathbf{x} = \{p \mid \mathbf{x} \in \operatorname{supp}(w_p)\}$. The blended global-deformation field is given by:

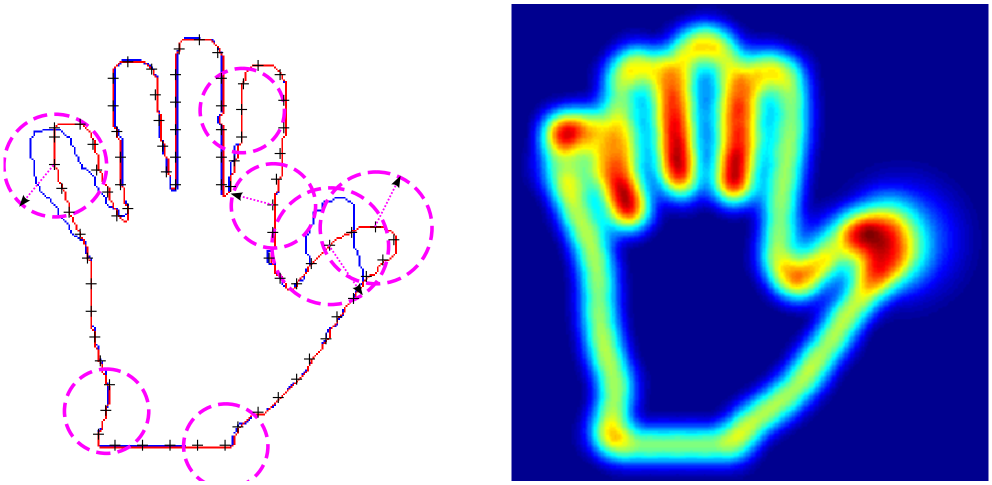
$$\mathbf{u}(\mathbf{x}) = \sum_{p \in N_\mathbf{x}} r_p(\mathbf{x})\mathbf{u}_p(\mathbf{x}), \qquad \text{with} \qquad r_p(\mathbf{x}) = \frac{w_p(\mathbf{x})}{\sum_{p' \in N_\mathbf{x}} w_{p'}(\mathbf{x})}. \tag{20}$$

Here, $r_p(\mathbf{x})$ ensures the partition-of-unity (PU), so that Equation 20 is essentially a weighted average. Compared to the B-spline model, the partition-of-unity method does not need explicit connections between neighboring patches, and allows for arbitrary overlapping of local models. This blending scheme has been previously used in the polyaffine model for diffeomorphic image registration [50], meshless image registration [11], and point-based computer graphics [8]. Some previous works [11, 9, 10] directly define the partition functions $r_p(\mathbf{x})$ without constructing them from weighting functions. However, weighting functions provide two appealing features.

First, weighting functions are intuitive to handle, they allow us to assign different weights and scales to patches, and can potentially be made adaptive according to prior information of shape deformation[4]. It is desirable that the positions, scales, and weighting factors of patches can be optimized according to registration results in a data-driven fashion. However, data-driven approaches are often application dependent, so this topic is beyond the scope of our paper. Here, we exploit a simple heuristic to register shape contours. Broadly speaking, we choose the patches' scales according to the shapes' mutual distance, since edge points that are far apart are more likely to have large-scale deformations. We found that this simple heuristic works well in most cases. Figure 7(a) shows an example shape with patches distributed along its contour. Figure 7(b) shows the sum of their weighting functions (i.e., the denominator of partition functions r_p in Equation 20). Thus, we limit the computation of the deformation model to take place along shape contours, and achieve a similar effect as the narrow-band function used in [5, 4].

Secondly, weighting functions also provide a way to specify our confidence about the approximation accuracy of different local deformation models. This confidence measure will be used to regularize the global deformation field. Regularizers are important for preventing degenerating solutions, since the solutions of shape registration may not be unique. Regularizers remove ambiguous solutions by penalizing undesired fluctuations in the estimated de-

[4] In the partition-of-unity literature [9], other a priori knowledge can also be included about local behavior of the solution in the finite element space.



(a) (b)

Figure 7: 2-D meshless deformation model for shape registration. a) Circular patches are distributed along the shape contours with centers marked as black crosses, and their scales are adapted according to distance of the source (in red) to the target shape (in blue). b) The sum of weighting functions, whose support covers the interested image region.

formation field, and prefer "smooth" solutions. Next, we study the problem of how to regularize a meshless shape-deformation model.

5.3. Smoothness regularizer

We have shown that a global deformation field relating two shapes can be obtained by blending local deformation fields using Equation 20. A classic regularizer used in B-spline models [5] is to penalize the magnitude of derivatives of the resulting deformation. However, calculating derivatives of our meshless model is not straightforward. For example, according to the product rule, first-order derivatives of the global deformation are obtained as follows:

$$\nabla \mathbf{u}(\mathbf{x}) = \sum_{p \in N_{\mathbf{x}}} \left(\nabla r_p(\mathbf{x}) \mathbf{u}_p(\mathbf{x}) + r_p(\mathbf{x}) \nabla \mathbf{u}_p(\mathbf{x}) \right). \tag{21}$$

The above differentiation involves derivatives of local deformation models $\mathbf{u}_p(\mathbf{x})$, and those of the partition functions $r_p(\mathbf{x})$ given by:

$$\nabla r_p(\mathbf{x}) = \frac{\nabla w_p(\mathbf{x}) \sum_{p' \in N_{\mathbf{x}}} w_{p'}(\mathbf{x}) - w_p(\mathbf{x}) \sum_{p' \in N_{\mathbf{x}}} \nabla w_{p'}(\mathbf{x})}{\left(\sum_{p' \in N_{\mathbf{x}}} w_{p'}(\mathbf{x}) \right)^2}. \tag{22}$$

It is clear that calculation of $\nabla \mathbf{u}(\mathbf{x})$ is complicated by the presence of blending functions. This problem was partially addressed in [11], by approximating $\nabla \mathbf{u}(\mathbf{x})$ using the differentiations of local models (i.e., $\nabla \mathbf{u}_p(\mathbf{x})$) as follows:

$$\widetilde{\nabla} \mathbf{u}(\mathbf{x}) = \sum_{p \in N_{\mathbf{x}}} r_p(\mathbf{x}) \nabla \mathbf{u}_p(\mathbf{x}), \tag{23}$$

that is basically a weighted average of the derivatives of neighboring patches. The approximated derivative $\widetilde{\nabla} \mathbf{u}(\mathbf{x})$ is then penalized in a similar fashion as in the classic smoothness regularizer [5]. However, this approximation is only accurate when neighboring patches have very similar derivatives (in conformity). This condition is achieved by penalizing a Sobolev conformity term between neighboring local deformation models as follows [11]:

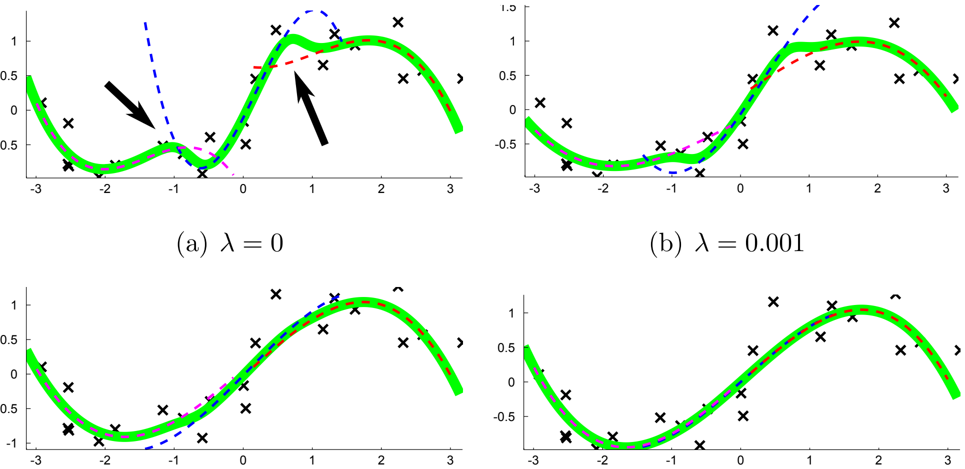
$$\mathbf{S}_k^{p,q} = \sum_{|\eta| \leq k} \int w_p(\mathbf{x}) w_q(\mathbf{x}) \| \mathbf{D}^{\eta} \mathbf{u}_p(\mathbf{x}) - \mathbf{D}^{\eta} \mathbf{u}_p(\mathbf{x}) \|_2^2 d\mathbf{x}, \tag{24}$$

where $\mathbf{D}^{\eta} \mathbf{u}_p(\mathbf{x}) = \partial_x^{\eta_x} \partial_y^{\eta_y} \mathbf{u}_p(\mathbf{x})$, $\eta_x, \eta_y = 0, 1, \ldots, k$, and $\eta = (\eta_x, \eta_y)$ is a multi-index vector with total order $|\eta| = \eta_x + \eta_y$. The usage of multi-index η is to include various orders of derivatives. In other words, the Sobolev conformity term is essentially a weighted squared sum of difference (SSD) between the derivatives of neighboring polynomial models. By enforcing $\mathbf{S}_k^{p,q} \to 0$ for each pair of patches (p,q), the local deformation models are forced to have similar derivatives. Equation 24 can be simplified into a quadratic term of local polynomial coefficients, but this approach is still much more involved than regularizing mesh models, and it involves two components: the Sobolev conformity term and a classic gradient-based smoothness term.

We found that a much simpler regularizer can be derived for smoothing the deformation field. First, the conformity between local models can be simply enforced by shifting and comparing the coefficients of local polynomial representations, instead of using an integral SSD form as in Equation 24. In this way, the conformity regularizer is greatly simplified, without having to compare the derivatives of local models. To differentiate with the conformity regularizer, we call our simplified functional as *consistency regularizer*, as it penalizes the inconsistencies in the polynomial coefficients of neighboring patches, instead of measuring the conformity of their derivatives. Secondly, our consistency term also penalizes fluctuations in the deformation field, and is enough to ensure its smoothness, further simplifying the registration framework by replacing the classic derivative-based regularizer.

In Figure 8, we show the idea of our consistency regularizer using a simple 1-D example of approximating scattered data points. Intuitively, the fluctuations in the global model is related to the inconsistencies among local deformation models (Figure 8(a)). One may consider that consistency between two local deformation fields, $\mathbf{u}_p$ and $\mathbf{u}_q$, can be measured as the Euclidean distance between their parameters $\mathbf{d}_p$ and $\mathbf{d}_q$ as follows:

$$E_{p,q} = \left\|\mathrm{vec}(\mathbf{d}_p - \mathbf{d}_q)\right\|^2 . \tag{25}$$



(a) $\lambda = 0$

(b) $\lambda = 0.001$

(c) $\lambda = 0.05$

(d) $\lambda = 1.0$

Figure 8: Consistency regularizer for 1-D scatter-point approximation. Data points were generated from sinusoid function with added noise. Three patches of equal radius and weighting factors were used to partition the computational domain $[-\pi, \pi]$. The parameter λ controls the relative contribution of the regularizer term. a) Fluctuations in the global model are related to inconsistencies between local polynomial models. b–d) By increasingly penalizing inconsistencies among local models, the global model becomes smoother.

Here, we stretched the parameter matrices $\mathbf{d}_p$, $\mathbf{d}_q$ into vectors $\text{vec}(\mathbf{d}_p - \mathbf{d}_q)$. In fact, a similar idea has been used for smoothing B-splines [17] by penalizing differences of adjacent B-splines' parameters. However, for meshless models, the consistency regularizer based on (25) incorrectly compares local deformation coefficients from different local coordinate systems. We demonstrate this effect using another 1-D example as shown in Figure 9. In this example, three linear (i.e., first-order polynomial) deformation models are blended into a global one. Even though the local models are themselves consistent, their coefficients are different due to different coordinate systems. The regularizer

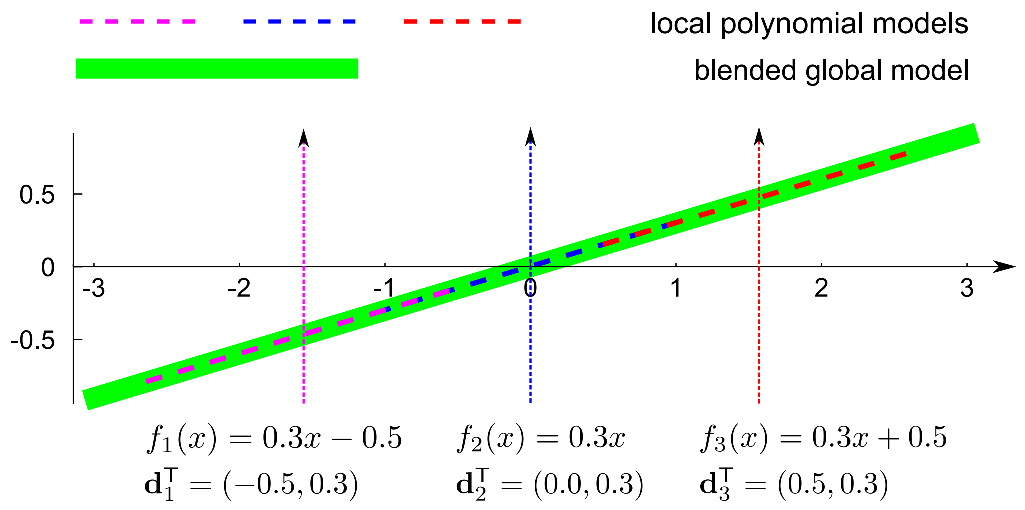


Figure 9: A 1-D example of shifting local coordinate systems. Three local models are blended into a global one. The local models are represented using first-order polynomials, with basis functions $\boldsymbol{\phi}(\mathbf{x}) = (1, x)^\mathsf{T}$. Shifting the coordinate system leads to a linear transform of local model coefficients.

in Equation 25 failed to take the coordinate transform into consideration. To address this issue, we need to shift and align the local coordinate systems so that $\mathbf{d}_p$ and $\mathbf{d}_q$ is compared on a common ground.

Fortunately, aligning the local coordinate systems can be achieved using a linear operator. To begin with, the representation of $\mathbf{u}_q(\mathbf{x})$ in the coordinate system of another patch p can be obtained by shifting $\mathbf{u}_q(\mathbf{x})$ as follows:

$$\begin{aligned} \mathbf{u}_q(\mathbf{x}) &= {\mathbf{d}_q}^\mathsf{T} \boldsymbol{\phi}_q(\mathbf{x}) \\ &= {\mathbf{d}_q}^\mathsf{T} \boldsymbol{\phi}(\mathbf{x} - \mathbf{q}) \\ &= {\mathbf{d}_q}^\mathsf{T} \boldsymbol{\phi}(\mathbf{x} - \mathbf{p} + \mathbf{p} - \mathbf{q}) \\ &= {\mathbf{d}_q}^\mathsf{T} \boldsymbol{\phi}_p(\mathbf{x} + \underbrace{\mathbf{p} - \mathbf{q}}_{\boldsymbol{\Delta}\mathbf{x}}) \end{aligned} \tag{26}$$

As a result, the alignment of local deformation models reduces itself to the shifting of the basis functions $\boldsymbol{\phi}_p(\mathbf{x})$. Our choice of monomial basis functions makes this computation particularly simple. In general, shifting a polynomial basis $\boldsymbol{\phi}(\mathbf{x})$ by $\boldsymbol{\Delta}\mathbf{x} = (\delta x, \delta y)$ leads to:

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}) &= (1, x + \delta x, y + \delta y, (x + \delta x)(y + \delta y), \ldots, (y + \delta y)^m)^{\mathsf{T}} \\ &= \mathbf{S}^{\mathsf{T}}(\boldsymbol{\Delta}\mathbf{x})\boldsymbol{\phi}(\mathbf{x}), \end{aligned} \tag{27}$$

where $\mathbf{S}^{\mathsf{T}}(\boldsymbol{\Delta}\mathbf{x})$ is the linear *basis-shifting-operator*. For example, for second-order basis vector $\boldsymbol{\phi}(\mathbf{x}) = (1, x, y, xy, x^2, y^2)^{\mathsf{T}}$, its shifting-operator is:

$$\mathbf{S}^{\mathsf{T}}(\boldsymbol{\Delta}\mathbf{x}) = \begin{pmatrix} 1 & \delta x & \delta y & \delta x \delta y & \delta^2 x & \delta^2 y \\ 0 & 1 & 0 & \delta y & 2\delta x & 0 \\ 0 & 0 & 1 & \delta x & 0 & 2\delta y \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{28}$$

By substituting (27) into (26), we obtain the shifted local deformation field:

$$\begin{aligned} \mathbf{u}_{q \to p}(\mathbf{x}) &= \mathbf{d}_q{}^{\mathsf{T}}\mathbf{S}^{\mathsf{T}}(\boldsymbol{\Delta}\mathbf{x})\boldsymbol{\phi}_p(\mathbf{x}) \\ &= (\mathbf{S}(\boldsymbol{\Delta}\mathbf{x})\mathbf{d}_q)^{\mathsf{T}}\boldsymbol{\phi}_p(\mathbf{x}). \end{aligned} \tag{29}$$

Here, we used $\mathbf{u}_{q \to p}(\mathbf{x})$ to distinguish $\mathbf{u}_q$ from its shifted representation. Therefore, shifting the local coordinate system leads to linearly transformed polynomial coefficients $\mathbf{d}_{q \to p} = \mathbf{S}(\boldsymbol{\Delta}\mathbf{x})\mathbf{d}_q$. Now, we can modify the consistency term in Equation 25 to include the shifting operator as:

$$E^c_{p,q} = \|\mathrm{vec}(\mathbf{d}_p - \mathbf{S}(\mathbf{p} - \mathbf{q})\mathbf{d}_q)\|^2 . \tag{30}$$

Finally, we can sum up the consistency term for every pair of neighboring patches as follows:

$$E^c = \frac{1}{N} \sum_p \left(\sum_q w_q(\|\mathbf{p} - \mathbf{q}\|) E^c_{p,q} \right), \tag{31}$$

where N is the number of patches, and the terms $E^c_{p,q}$ are weighted by w_q based on the distance between patches. The inclusion of weighting functions has two purposes. First, it effectively limits the interaction range of a patch p to its neighboring patches q that contain the center of p, otherwise $w_q(\|\mathbf{p} - \mathbf{q}\|) = 0$. Secondly, the weighting function gives more emphasis to the consistency of neighboring patches that are close to each other.

5.4. Verification of the regularizer

We have introduced a simple consistency regularizer based on the linear shifting operator in (31). Before adopting it for shape registration, we would like to briefly verify its ability of smoothing the deformation fields. We have shown in Figure 8 an example of 1-D approximation of scattered data points, combining the consistency regularizer with a SSD data-fitting cost function. Formally, the combined data-fitting functional is given by:

$$E_{\text{approx}} = \sum_i \left(\mathbf{u}(\mathbf{x}) - f_i(\mathbf{x}_i)\right)^2 + \lambda E^c, \tag{32}$$

where $(\mathbf{x}_i, f_i(\mathbf{x}_i))$ are scattered data points, and λ is the weight of the consistency regularizer. Figure 8 illustrates how the increase of the regularizer's weight leads to smoother global approximation, and the global model approximates a polynomial fitting asymptotically. The effect of the consistency regularizer can also be analyzed theoretically. Here, we aim at providing an intuitive understanding of the regularizer rather than a rigorous proof.

On one hand, we can show that minimizing the consistency term in (25) can also enforce the conformity between local deformation models, i.e., the continuity in their derivatives. To verify, note that the differential operator on polynomials can again be represented as a linear operator on the coefficients $\mathbf{d}_p$. Let us denote a general differential operator and its matrix representation using the same symbol $\mathbf{D}^\eta$. Combining it with our shifting operator into the conformity term in Equation 24, we obtain:

$$\begin{aligned}
\mathbf{S}_k^{p,q} &= \sum_{|\eta|\leq k} \int w_p(\mathbf{x}) w_q(\mathbf{x}) \left(\mathbf{D}^\eta \mathbf{u}_p(\mathbf{x}) - \mathbf{D}^\eta \mathbf{u}_{q\to p}(\mathbf{x})\right)^2 d\mathbf{x} \\
&= \sum_{|\eta|\leq k} \int w_p(\mathbf{x}) w_q(\mathbf{x}) \left(\mathbf{D}^\eta \mathbf{d}_p \boldsymbol{\phi}_p(\mathbf{x}) - \mathbf{D}^\eta \mathbf{S}(\mathbf{p}-\mathbf{q}) \mathbf{d}_q \boldsymbol{\phi}_p(\mathbf{x})\right)^2 d\mathbf{x} \\
&\leq \|\mathrm{vec}\left(\mathbf{d}_p - \mathbf{S}(\mathbf{p}-\mathbf{q})\mathbf{d}_q\right)\|^2 \sum_{|\eta|\leq k} \|\mathbf{D}^\eta\|^2 \int w_p(\mathbf{x}) w_q(\mathbf{x}) \boldsymbol{\phi}_p^2(\mathbf{x}) d\mathbf{x}.
\end{aligned} \tag{33}$$

Consequently, the conformity term is upper-bounded by our consistency regularizer, and minimizing the consistency regularizer will similarly enforce the conformity condition among local models.

On the other hand, when $E^c \to 0$, the blended global model is coerced into a polynomial one, and the order of this polynomial is the same as the basis functions. For instance, if we choose the linear basis vector $\boldsymbol{\phi} = (1, x, y)^\mathsf{T}$, then $E^c = 0$ implies that $\mathbf{u}(\mathbf{x})$ is a piecewise linear function and $\|\nabla \mathbf{u}_x\|^2 + \|\nabla \mathbf{u}_y\|^2 = 0$. As a result, our consistency regularizer has the same asymptotic property as the classic gradient-based regularizers.

We have shown the relationship of our consistency regularizer with both the conformity term used in a recent partition-of-unity method [11] and the gradient-based regularizers for mesh models [5]. Next, we formulate and solve the nonrigid shape registration problem, by combining the meshless deformation model with our variational chamfer-matching energy.

6. Registration using gradient-descent method

With the meshless deformation model, the registration problem is now converted to the one of seeking the "best" deformation parameters $\mathbf{d}_p$, $p \in \mathbb{P}$, by combining both the variational chamfer-matching and the consistency regularizer. The chamfer-matching energy measures the closeness of registered shape contours (data term), and the consistency regularizer penalizes fluctuations in the deformation field (smoothness term). Formally, we formulate nonrigid shape registration as the following functional minimization problem:

$$\mathbf{d}_p = \arg\min_{\mathbf{d}_p} E^v = \arg\min_{\mathbf{d}_p} \left(E^d(\mathbf{u}) + \lambda E^c\right), \tag{34}$$

where parameter λ defines the relative importance of the consistency term. As in previous works [5, 2], minimizing (34) can be achieved using gradient descent. The gradients can be derived as follows given by:

$$\frac{\partial E^v}{\partial \mathbf{d}_p} = \frac{\partial E^d(\mathbf{u})}{\partial \mathbf{d}_p} + \lambda \frac{\partial E^c}{\partial \mathbf{d}_p}. \tag{35}$$

Gradients of the consistency energy $\frac{\partial E^c}{\partial \mathbf{d}_p}$ can be calculated from (31) as:

$$\begin{aligned} \frac{\partial E^c}{\partial \mathbf{d}_p} &= \frac{1}{N} \sum_q w_q(\|\mathbf{p} - \mathbf{q}\|) \frac{\partial E^c_{p,q}}{\partial \mathbf{d}_p} \\ &= \frac{1}{N} \sum_q w_q(\|\mathbf{p} - \mathbf{q}\|) \left(\mathbf{d}_p - \mathbf{S}(\mathbf{p} - \mathbf{q})\mathbf{d}_q\right). \end{aligned} \tag{36}$$

Gradients of the variational chamfer-matching energy can be calculated from chain-rule of variational calculus, to obtain:

$$\begin{aligned} \frac{\partial E^d(\mathbf{u})}{\partial \mathbf{d}_p} &= \int \mathbf{J}(\mathbf{x}) \cdot \frac{\partial \mathbf{u}(\mathbf{x})}{\partial \mathbf{d}_p} d\mathbf{x} \\ &\approx \int \widetilde{\mathbf{J}}(\mathbf{x}) \cdot r_p(\mathbf{x}) \boldsymbol{\phi}(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{37}$$

Algorithm 1: Shape registration based on gradient-descent.

```
Input: Binary maps for source and target shapes ($\mathcal{S}$ and $\mathcal{D}$),
       partition-of-unity patches $\mathbb{P}$, consistency weighting factor $\lambda$.
Output: Deformation field $\mathbf{u}(\mathbf{x})$.
Initialize local deformation coefficients $\mathbf{d}_p^0 \leftarrow 0$, for $p \in \mathbb{P}$;
Calculate distance map of the target shape $\Pi_{\mathcal{D}}$;
while Not converge do
    Calculate global deformation $\mathbf{u}^k(\mathbf{x})$ from local deformation
    coefficients $\mathbf{d}_p^k$ (Equation 20);
    Warp source shape as $\mathcal{S}^k(\mathbf{x}) = \mathcal{S}(\mathbf{x} + \mathbf{u}^k)$ ;
    Calculate distance map $\Pi_{\mathcal{S}^k(\mathbf{x})}$;
    Update Jacobian matrix (Equation 36 and Equation 37);
    Update $\mathbf{d}_p^{k+1}$ using gradient-descent rules;
end
```

Here, $\mathbf{J}(\mathbf{x})$ is left-hand-side of the Euler-Lagrange equation of $E^d(\mathbf{u})$, and we use its approximation $\widetilde{\mathbf{J}}(\mathbf{x})$ defined in Equation 16.

Using the derived gradients, we implemented an optimization algorithm based on the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [51]. At each iteration of the algorithm, the source shape is first warped using the deformation field reconstructed from local field parameters (Equation 20), and then its distance transform $\Pi_{\mathcal{S}(\mathbf{x}+\mathbf{u})}$ is updated. Both the destination shape $\mathcal{D}$ and its distance transform $\Pi_{\mathcal{D}}$ remain constant. Conceptually, the gradient-descent process is described in Algorithm 1.

Direct warping of a shape contour might cause deterioration of image quality since the deformation $\mathcal{S}(\mathbf{x}+\mathbf{u})$ is often implemented using interpolated backward mapping. Evidently, a warped 0–1 edge map may produce decimal pixel values, i.e., no longer a binary image. In practice, the deterioration of edge maps is rarely an issue for our algorithm, and can be naturally handled by our variational chamfer-matching gradient (Equation 16), by relaxing the binary edge map to a gray-scale one. Furthermore, we can use an extended version of the distance transform defined on gray-scale images that takes the gray-scale edge map value into account. Nevertheless, in this paper, we want to keep the edge map binary, for better evaluation of the registration results, since the compared methods in [5, 1] are based on binary representations.

In previous implementation of distance-based registration [5] [5], the shape contours are represented as connected line segments, that is basically a vectorized representation. Warping of the shape contours is achieved by deforming the end points of these line segments. Thus, interpolation and resulting deterioration of edge maps can be avoided. This approach may be adopted for our method, but is not straightforward to implement. Fortunately, in many applications, the input shape contours are detected from grayscale source images. In these cases, we simply warp the source images, and then apply an edge-detection step to recover the deformed binary contour. To summarize, we modify the registration Algorithm 1 to obtain an edge-preserving one given by Algorithm 2 in which the main differences are highlighted in italic. Both the distance transform and edge detection are highly efficient operations, with minimal influence on the algorithm's computational cost.

[5] Available for download at `http://www.cse.lehigh.edu/~huang/downloads.htm`

Algorithm 2: Gradient-descent with edge-preserving.

```
Input: Source and target images ($I_s$ and $I_t$), partition-of-unity
       patches $\mathbb{P}$, consistency weighting factor $\lambda$.
Output: Deformation field $\mathbf{u}(\mathbf{x})$.
Initialize local deformation coefficients $\mathbf{d}_p^0 \leftarrow 0$, for $p \in \mathbb{P}$;
Detect target shape contour from the image $\mathcal{D} \leftarrow I_t$;
Calculate distance map of the target shape $\Pi_{\mathcal{D}}$;
while Not converge do
    Calculate global deformation $\mathbf{u}^k(\mathbf{x})$ from local deformation
    coefficients $\mathbf{d}_p^k$ (Equation 20);
    Warp source image as $I_s^k(\mathbf{x}) = I_s(\mathbf{x} + \mathbf{u}^k)$ ;
    Detect source shape contour $\mathcal{S}^k \leftarrow I_s^k$ ;
    Calculate distance map $\Pi_{\mathcal{S}^k(\mathbf{x})}$;
    Update Jacobian matrix (Equation 36 and Equation 37);
    Update $\mathbf{d}_p^{k+1}$ using gradient-descent rules;
end
```

Finally, we adopt the hierarchical multi-scale registration strategy used in previous nonrigid registration methods [5, 31, 11] in order to avoid local minima (i.e., a coarse-to-fine approach). Briefly, multi-scale registration starts with sub-sampled images (at the coarser level). The deformation obtained at coarse scale is then used to initialize the algorithm at finer scales.

7. Evaluation

In this section, we evaluate our method on a number of experiments, and compare it with the methods by Huang et al. [5] and Chen and Bhanu [1]. As described in Section 3, Huang's method is based on distance-map representation and a B-spline deformation model, while Chen and Bhanu's method is based on shape-context and a thin-plate deformation model. Our goal in this comparison is to demonstrate two aspects of our method. First, by removing the narrow-band function, our chamfer-matching functional can help improving the registration accuracy of distance-based methods at high-curvature regions. Secondly, by replacing spline-based mesh models with our partition-of-unity model, we can handle challenging shape deformations that may be difficult for existing methods. We tested our method on the Brown University shape dataset [52], that was also used by many previous methods [5, 1, 4]. Additionally, we demonstrate an application of our method to image sequences of cell mitosis.

7.1. Brown University shape dataset

Our method requires some parameters to be set. First, we empirically set the consistency regularizer's weighting factor $\lambda = 0.001$. Secondly, the representation power of our meshless model depends on both the density and distribution the image domain partitions. To simplify, we began with regularly distributed patches with radius $r = 20$ (pixels) and the spaces between patches $d = 6$. We normalized the images of Brown University shape dataset to sizes 150×150 pixels. Finally, we globally aligned the shapes using the rigid registration method implemented in [53]. Figure 10 shows registration results produced using our method. For a quantitative evaluation, we used the average mutual distance between the registered shape contours [1]. For qualitative comparison, we also visualize the results in a similar manner as

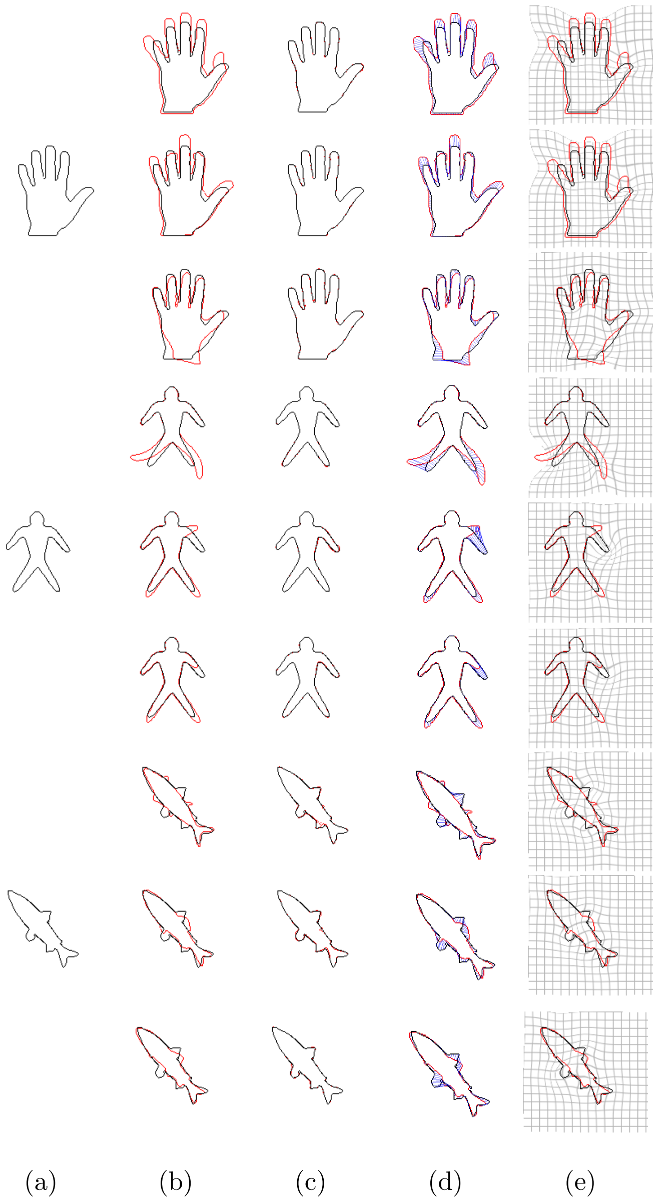


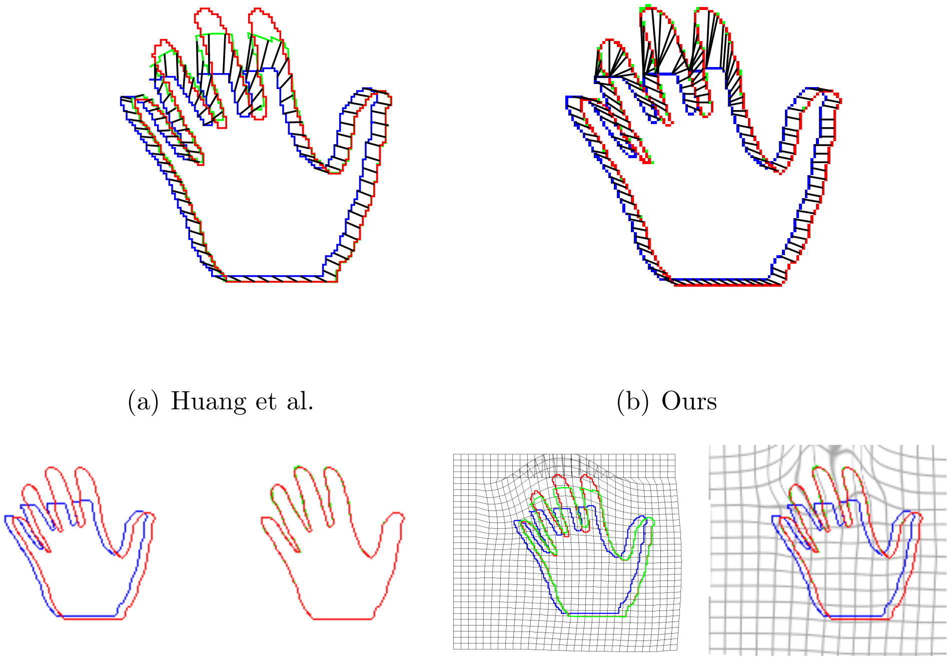
Figure 10: Brown university shape dataset. (a) Target images. (b) Overlaid target (in black) and source images (in red) before registration. (c) After registration. (d) Correspondence between target and source images. (e) Deformation fields as distorted grids.

in [5, 4, 1]. As in [1], we selected three different shapes (person[6], hand, and fish). The average pixel distances after local registration for person, fish, and hand were 0.14, 0.24, and 0.08, respectively. This result was better than the one reported in [1] (0.59, 0.45 and 0.61, respectively). Additionally, for most cases, the maximum pixel distance was less than 2 pixels showing that registration quality was consistent along contours. We believe that the use of both the chamfer-matching functional and meshless model contributed to the improved accuracy. On one hand, chamfer-matching functional is a more direct measurement of shape contour's alignment than shape-context descriptors. On the other hand, the partition-of-unity model is numerically more accurate than thin-plate splines that use radial-basis functions.

7.1.1. Shapes with occlusion and large deformation

Our method was able to register shapes with occlusion, large deformation, and high curvature. Here, we compared our method with Huang's method [5]. Figure 11 shows the registration of occluded hand shapes. In [5], landmarks were placed at occluded parts to guide the registration process. Without landmarks, it is difficult for splines to handle large deformation from the source (blue) to the target (red) shape due to foldings in the mesh model (Figure 11(e)). In contrast, our meshless model alleviated the folding effect (Figure 11(f)). Here, we plotted the deformation field by deforming an image of grid, but it does not mean that our model is restricted to such a topology. Quantitatively, our method still achieved remarkable accuracy with the mean, maximum, and variance of edge distances as 0.15, 1.41 and 0.13, respectively,

[6] Named dude in the original dataset.


(a) Huang et al.

(b) Ours

(c) Before

(d) After

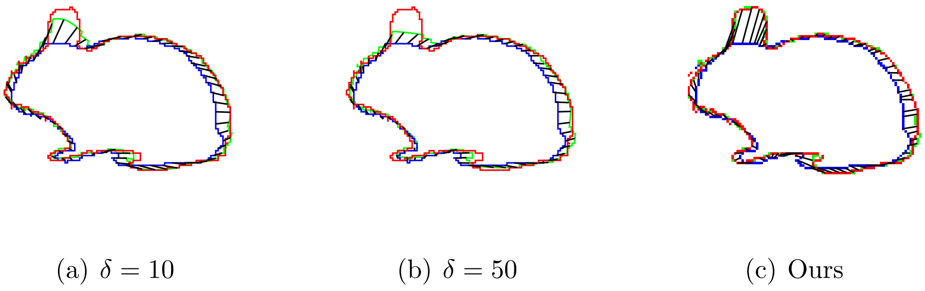
(e) Huang et al.

(f) Our method

Figure 11: Occluded shapes (hand). The contours are source shape (blue), target shape (red), and deformed source shape (green). a) Huang et al. [5], average distance: 0.64, max: 9.2, variance: 1.99 . b) Our method, average distance: 0.15, max: 1.41, variance: 0.13. c) Deformed source (green) and target (red) shapes. d) The deformation field represented using a deformed grid image.

in comparison with Huang's result (mean: 0.64, max: 9.2, variance: 1.99).

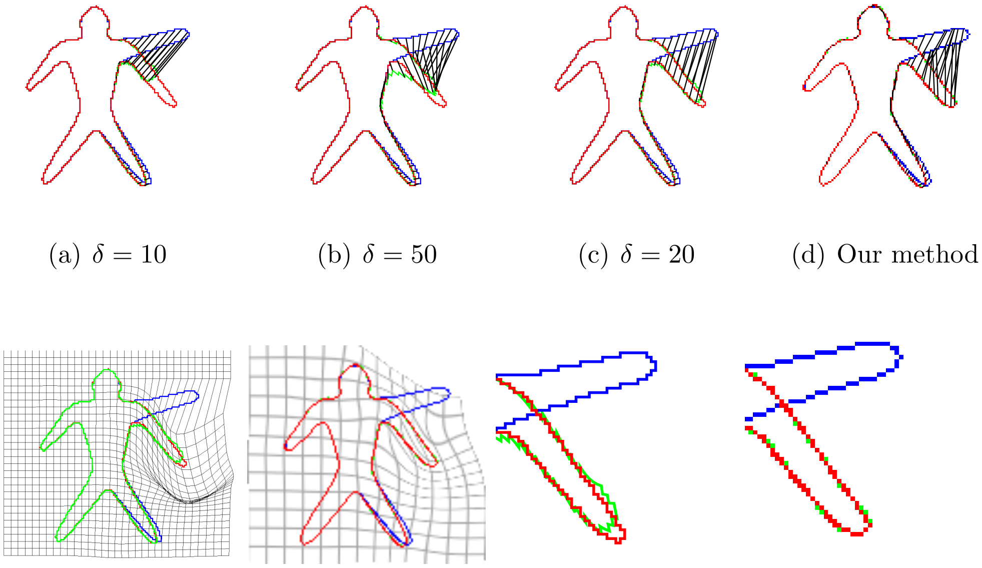
The mesh model's folding effect may be further complicated by the narrow-band function N_δ. Figure 12 shows registration of occluded shapes (bunny) using increasing narrow-band δ. On one hand, we observed that increasing δ will generally reduce the ability of Huang's method to handle high-curvature regions (Figure 12(a) and 12(b)) as using large δ may include embedding



(a) $\delta = 10$ (b) $\delta = 50$ (c) Ours

Figure 12: Occluded shapes (rabbit). a) Huang et al. [5], average distance: 0.62, maximum:5.38, variance: 0.89. b) Increasing the narrow-band width futher reduces the registration accuracy, with average distance: 1.00, maximum: 10.44, variance: 2.63. c) Our method, average distance: 0.41, maximum: 2.23, variance: 0.28.

space that undergoes high-curvature bending (see Figure 2 for demonstration), while our method was not affected at all by such a parameter (Figure 12(c)). On the other hand, with decreasing δ, Huang's method became prone to local minima. Figure 13 shows an example of shapes with both high curvature and large deformations. Here, we tested Huang's method using different narrow-band width δ. When δ was small (Figure 13(a)), the algorithm converged prematurely, unable to extend the deformation large enough. When δ was large (Figure 13(b)), the method was unable to cope with high-curvature regions around the bending arm. An optimal result was obtained around $\delta = 20$. However, even this optimal result was still less accurate when compared to our method.



(a) $\delta = 10$ (b) $\delta = 50$ (c) $\delta = 20$ (d) Our method

(e) Huang et al. (f) Our method (g) Huang et al. (h) Our method

Figure 13: Large deformation (dude). a-c) Method of Huang et al., with varying narrow-band width. Best result was obtained when $\delta = 20$, with average distance: 0.17, max: 3.61, variance: 0.20. d) Ours, average: 0.14, max: 1.00, variance: 0.12.

7.2. Adapted partition-of-unity

We now extend our comparison with Huang's method using heuristically adapted patches introduced in Section 5.2 in order to reduce the computation domain to be around the shape contours. To be specific, we sampled along the shape contour at an equal geodesic distance d (usually $5 \leq d \leq 20$), and placed a patch centered at each sample point with its scale set to be $r_p = \max\left(r_m, \kappa * \Pi_{\mathcal{D}(\mathbf{x})}(\mathbf{p})\right)$, where $r_m = \rho\, d$ was the minimum radius so that the patches overlapped with one another $(1.5 \leq \rho \leq 3$, and $\kappa = 2)$. In other words, the patches' scale increased proportionally if the two shapes were far

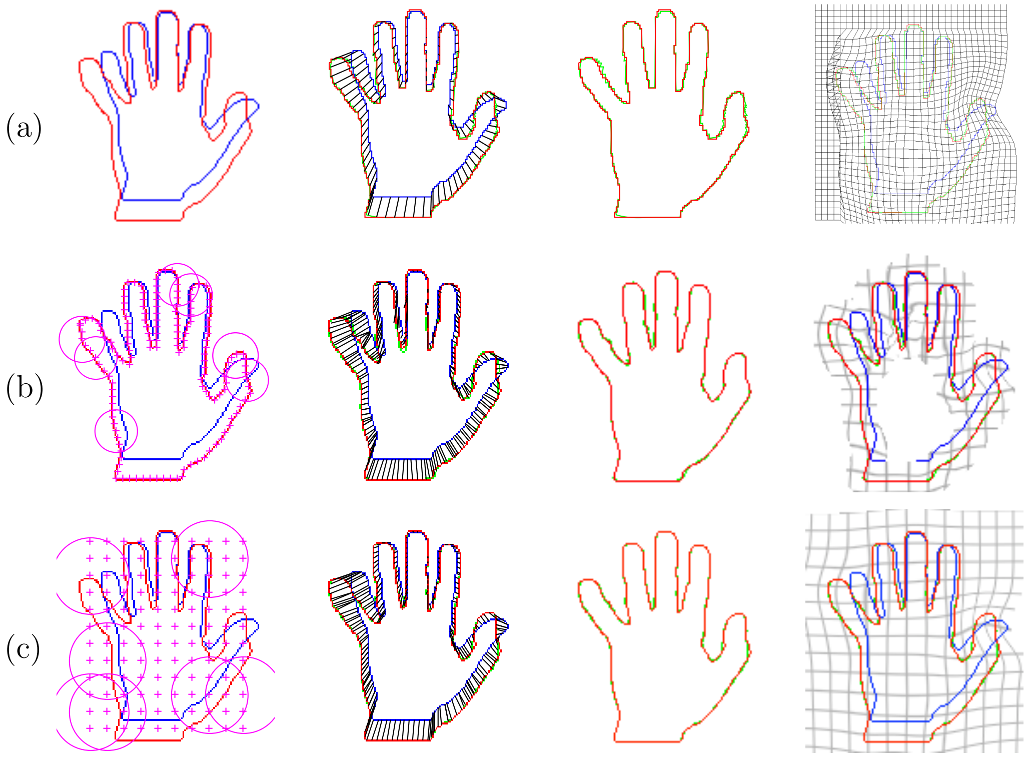


Figure 14: Registration using adaptive patches (hand). a) Results of Huang's method (mean: 0.29, max: 2.00, variance: 0.22). b) Adapted partitions and results (mean: 0.22, max: 2.00, variance: 0.19). g) Regularly distributed patches. c) Regular partitions and results (mean: 0.22, max: 2.00, variance: 0.18).

from each other. Figure 14, Figure 15 and Figure 16 show examples of shape registration using adapted patches. In all these examples, the adapted-patch method achieved results comparable to the ones using regular partitions and better results than Huang's method, while reducing the computational time by at least 50 percent. More reductions in computational cost are possible if we use finer partitions based on prior knowledge provided, for example, by statistical models of specific shapes.

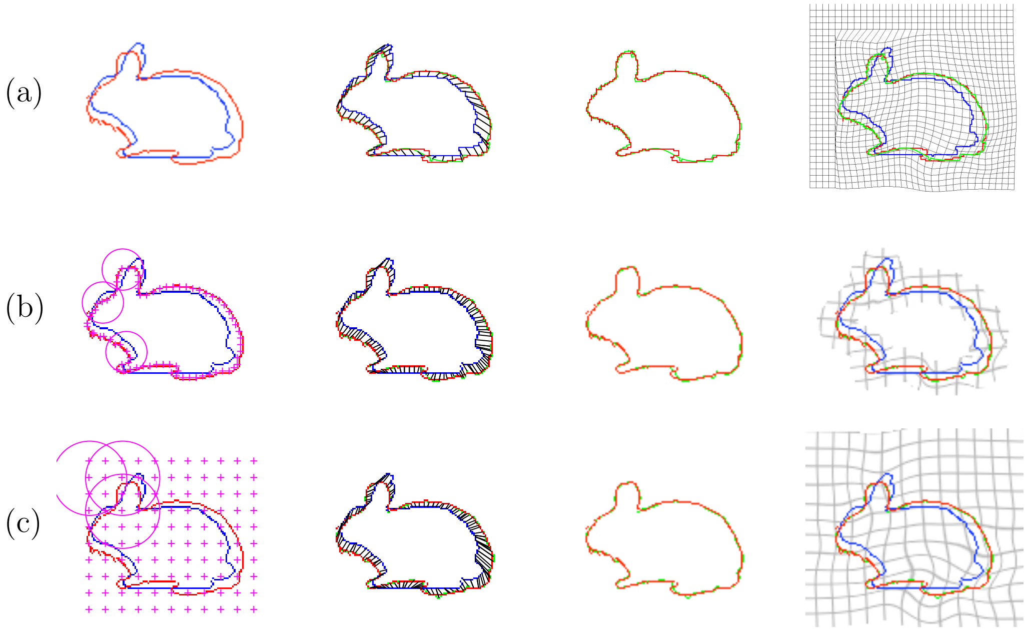


Figure 15: Adapted partitions (bunny). a) Results of Huang's method (mean: 0.48, max: 3.16, variance: 0.39). b) Adapted partitions and results (mean: 0.40, max: 2.83, variance: 0.30). g) Regular partitions and results (mean: 0.40, max: 2.83, variance: 0.29).

7.3. Cell division

Finally, we demonstrate an application of our method on analyzing cell deformation. Tracking cell contour is of great interest in biomedical image analysis. Many existing methods are based on active contours and level-set representations. Although these methods produce good results in segmenting cell contours, few of them provide correspondences between deforming contours. Our method can be combined with level-set segmentation algorithms to further establish these correspondences. Figure 17 shows a sequence of cell mitosis. The top row shows the segmentation results obtained using the level-set segmentation method by Chunming Li et al. [33]. The second row

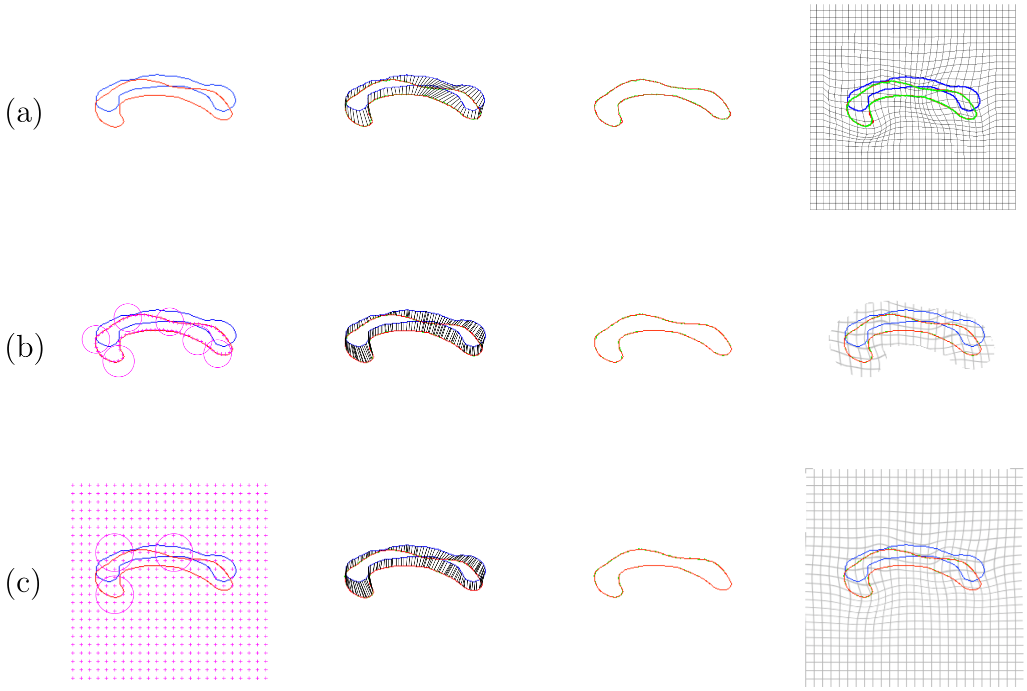


Figure 16: Adapted partitions (corpus callosum). a) Results of Huang's method (mean: 0.36, max: 2.0, variance: 0.25). b) Adapted partitions and results (mean: 0.25, max: 1.41, variance: 0.19). g) Regular partitions and results (mean: 0.21, max: 1.00, variance: 0.17).

shows overlaid shape contours from adjacent images. The third row shows the correspondence established using our method with adapted partition-of-unity. The deformed source contour (green) is well aligned with the target (blue). Interestingly, at the moment of cell mitosis, shape topology changed but our method still established reasonable correspondence between shapes of different topologies, considering that no continuous deformation actually exists to change a shape's topology.

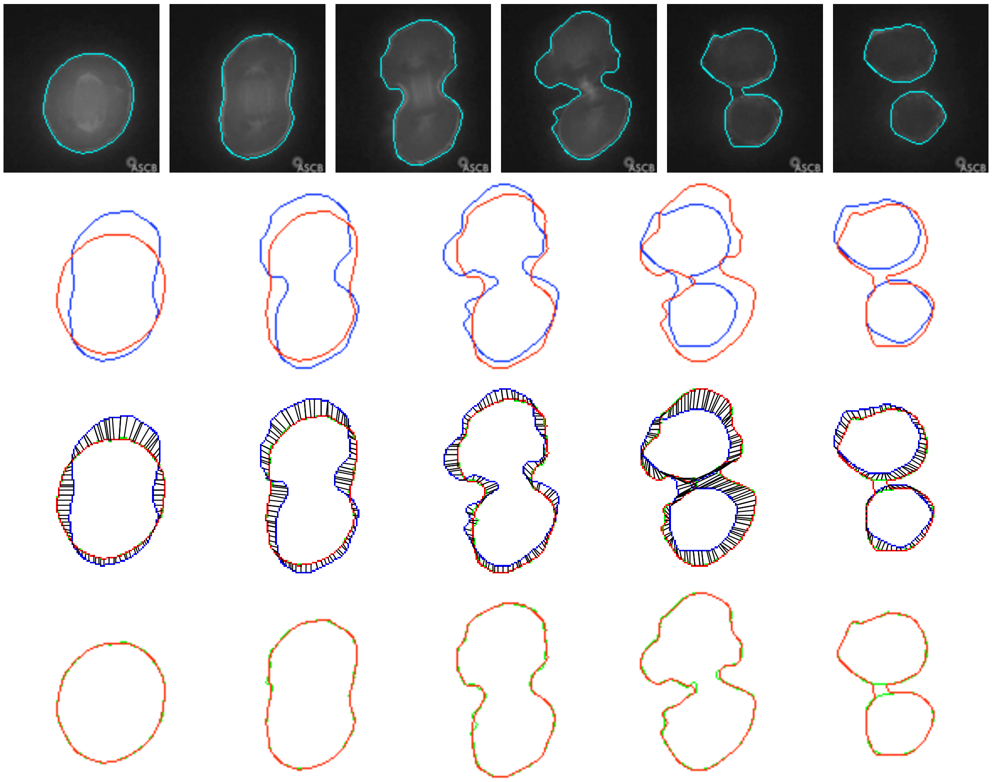


Figure 17: Cell-mitosis sequence. Row 1: Cell contours using level-set segmentation in [33]. Row 2: Overlaid source (blue) and target shapes (red). Row 3: Correspondence established using our method. Row 4: Target shape overlaid over the deformed source shape (green).

8. Conclusions and future work

A meshless nonrigid shape-registration algorithm was presented. The registration functional is a variational extension of the classic chamfer-matching energy in which distance transforms provide registration-error gradients, facilitating efficient registration. Also, we modeled shape deformation using a meshless parametric representation. This model does not rely on a regular control-point grid, and can be adapted to arbitrary shapes. Thus, registration can be focused around the shape contours, greatly improving computational efficiency. We tested the proposed method by registering a number of synthetic shapes, and a deforming cell sequence. Future work includes a 3-D extension of the method, and the handling of topological changes.

Despite promising results, our method still encounters problems in registering shapes that have large curvatures, and undergo high-degree deformation, causing local minima in the registration error. We believe that this problem can be addressed by adopting global-optimization algorithms such as simulated annealing [51], or by the inclusion of statistical priors [6].

References

[1] Hui Chen and Bir Bhanu. Global-to-local non-rigid shape registration. In *ICPR*, pages 57–60, Washington, DC, USA, 2006. IEEE Computer Society.

[2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 10(6):849–865, 1988.

[3] A. Thayananthan, B. Stenger, PHS Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *CVPR*, volume 1, 2003.

[4] N. Paragios, M. Rousson, and V. Ramesh. Non-rigid registration using distance functions. *Comp. Vision and Image Underst.*, 89(2-3):142–165, 2003.

[5] X. Huang, N. Paragios, and D.N. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 28(8):1303, 2006.

[6] M. Rousson and N. Paragios. Prior knowledge, level set representations & visual grouping. *Int. J. Comput. Vision*, 76(3):231–243, 2008.

[7] H.A. El Munim and A.A. Farag. Shape representation and registration using vector distance functions. In *CVPR*, pages 1–8, 2007.

[8] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.

[9] J. M. Melenk and I. Babuska. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289 – 314, 1996.

[10] G. R. Liu. *Mesh free methods: moving beyond the finite element method (Second Edition).* CRC, July 2009.

[11] S. Makram-Ebeid and O. Somphone. Non-rigid image registration using a hierarchical partition of unity finite element method. *ICCV*, 510:7, 2009.

[12] G.K. Rohde, A. Aldroubi, and B.M. Dawant. The adaptive bases algorithm for intensity-based nonrigid image registration. *IEEE Transactions on Medical Imaging*, 22(11):1470–1479, 2003.

[13] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.

[14] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, 2003.

[15] Yukie Nagai, Yutaka Ohtake, and Hiromasa Suzuki. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1339–1348, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[16] M.Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa. Fast directional chamfer matching. In *CVPR*, pages 1696–1703. IEEE, 2010.

[17] P.H.C. Eilers and B.D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–102, 1996.

[18] Wei Liu and Eraldo Ribeiro. A meshless method for variational nonrigid 2-d shape registration. In *ISVC 2010*, 2010.

[19] Wei Liu and Eraldo Ribeiro. A novel consistency regularizer for meshless nonrigid image registration. In *ISVC 2010*, 2010.

[20] P.F. Felzenszwalb and D.P. Huttenlocher. Distance transforms of sampled functions. *Cornell Computing and Information Science Technical Report TR2004-1963*, 2004.

[21] D. Gavrila. Pedestrian detection from a moving vehicle. In *Computer Vision ECCV 2000*, volume 1843, pages 37–49, 2000.

[22] Jamie Shotton, Andrew Blake, and Roberto Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30:1270–1281, July 2008.

[23] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. *Computer Vision–ECCV 2006*, pages 575–588, 2006.

[24] Jiayong Zhang, Robert Collins, and Yanxi Liu. Representation and matching of articulated shapes. In *CVPR*, pages 342–349, Washington, DC, USA, 2004. IEEE Computer Society.

[25] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid. From images to shape models for object detection. *International Journal of Computer Vision*, 87:284–303, 2010.

[26] X. Bai, X. Wang, L.J. Latecki, W. Liu, and Z. Tu. Active skeleton for non-rigid object detection. In *ICCV*, pages 575–582. IEEE, 2009.

[27] R.W. Put, E.M. Kerkhof, B.W. Raaymakers, IM Jürgenliemk-Schulz, and JJW Lagendijk. Contour propagation in MRI-guided radiotherapy treatment of cervical cancer: the accuracy of rigid, non-rigid and semi-automatic registrations. *Physics in medicine and biology*, 54:7135, 2009.

[28] Muayed S. Al-Huseiny, Sasan Mahmoodi, and Mark S. Nixon. Robust rigid shape registration method using a level set formulation. In *ISVC*, pages 252–261, Berlin, Heidelberg, 2010. Springer-Verlag.

[29] G. Tsechpenakis, L. Bianchi, D.N. Metaxas, and M. Driscoll. A novel computational approach for simultaneous tracking and feature extraction of C. elegans populations in fluid environments. *IEEE Transactions on Biomedical Engineering*, 55(5):1539–1549, 2008.

[30] N.A. Parra. *Rigid and non-rigid point-based medical image registration.* PhD thesis, Florida International University, 2009.

[31] M. Taron, N. Paragios, M.P. Jolly, É.C. de Paris, and F. CERTIS. Registration with uncertainties and statistical modeling of shapes with variable metric kernels. *IEEE Trans. on patt. anal. and mach. intell.*, 31(1):99–113, 2009.

[32] Chenyang Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359 –369, March 1998.

[33] C. Li, C. Xu, C. Gui, and M. D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Trans. Image. Process.*, 19(12):3243–3254, December 2010.

[34] N. Paragios. A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Transactions on Medical Imaging*, 22(6):773–776, 2003.

[35] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[36] M.S. Hansen, R. Larsen, D. Lyngby, B. Glocker, N. Navab, and G. Múnchen. Adaptive parametrization of multivariate B-splines for image registration. In *CVPR*, pages 1–8, 2008.

[37] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, 2003.

[38] Daniel Rueckert, Paul Aljabar, Rolf Heckemann, Joseph Hajnal, and Alexander Hammers. Diffeomorphic registration using b-splines. In *Medical Image Computing and Computer-Assisted Intervention*, volume 4191, pages 702–709. Springer, 2006.

[39] Wei Liu and Eraldo Ribeiro. A survey on image-based continuum-body motion estimation. *Image and Vision Computing*, In Press, Accepted Manuscript:–, 2011.

[40] H. Liu and P. Shi. Meshfree particle method. In *ICCV*, volume 1, pages 289–269, October 2003.

[41] T. Chen, X. Wang, D. Metaxas, and L. Axel. Fast motion tracking of tagged MRI using angle-preserving meshless registration. In *MICCAI*, page 320, 2008.

[42] T. Chen, S. Kim, S. Goyal, S. Jabbour, J. Zhou, G. Rajagopal, B. Haffty, and N. Yue. Object-constrained meshless deformable algorithm for high speed 3D nonrigid registration between CT and CBCT. *Medical physics*, 37:197, 2010.

[43] D. Sỳkora, J. Dingliana, and S. Collins. As-rigid-as-possible image registration for hand-drawn cartoon animations. In *International Symposium on Non-Photorealistic Animation and Rendering*, pages 25–33. ACM, 2009.

[44] X. Wang, D. Metaxas, T. Chen, and L. Axel. Meshless deformable models for LV motion analysis. In *CVPR*, pages 1–8, 2008.

[45] H.P. Ho, Y. Chen, H. Liu, and P. Shi. Point-based geometric deformable models for medical image segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 278–285, 2005.

[46] T.Y. Lee and S.H. Lai. 3D non-rigid registration for MPU implicit surfaces. In *CVPR Workshops*, pages 1–8. IEEE, 2008.

[47] C. Walder, M. Breidt, H. Bülthoff, B. Schölkopf, and C. Curio. Markerless 3D face tracking. *Pattern Recognition*, pages 41–50, 2009.

[48] G. Sapiro. *Geometric partial differential equations and image analysis.* Cambridge Univ Pr, 2001.

[49] I.M. Gel'fand, G.E. Shilov, and E. Saletan. *Generalized functions*, volume 1. Academic Press New York, 1964.

[50] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and simple calculus on tensors in the Log-Euclidean framework. In *MICCAI*, pages 115–122. Springer, 2005.

[51] J.F. Bonnans and C. Lemaréchal. *Numerical optimization: theoretical and practical aspects.* Springer-Verlag New York Inc, 2006.

[52] Daniel Sharvit, Jacky Chan, Hüseyin Tek, and Benjamin B. Kimia. Symmetry-based indexing of image databases. *J. of Vis. Comm. and Image Repres.*, 9(4):366–380, December 1998.

[53] Dirk-Jan Kroon and Cornelis H. Slump. MRI modality transformation in demon registration. In *ISBI*, pages 963–966. IEEE, 2009.